GADI (Genesis Aided Dispatch Interface)
Software Version 2.4.5
User Manual

GADI

Genesis Aided Dispatch Interface

800-2.2.0-A.1
January 2015

Printed in the U.S.A.

Trademarks

The following are registered trademarks of Motorola: SmartZone, SmartNet, Astro.

Any other brand or product names are trademarks or registered trademarks of their respective holders.

The Genesis Group Trademark Information

GenWatch3® is a registered trademark of GenCore Candeo, LTD., a subsidiary of Burks Gencore Co., Inc. D.B.A. The Genesis Group and Phil Burks.

Copyright

Includes technology licensed from Motorola.

Disclaimer

The GADI Users Manual is printed in the U.S.A. Burks Gencore Co., Inc. D.B.A. The Genesis Group and Phil Burks believe that the information included in this manual is correct; however, Burks Gencore Co., Inc. D.B.A. The Genesis Group and Phil Burks reserves the right to alter, revise and make periodic changes to the manual and its contents. Burks Gencore Co., Inc. D.B.A. The Genesis Group does not assume responsibility to notify any person of such revisions or changes. While we have taken strides to carefully examine our software and documentation and believe that it is reliable, the Genesis Group and Phil Burks assume no responsibility for the use of the manual, or GADI software, nor for any patent infringements or other rights of third parties who may use the manual or the GADI software. Burks Gencore Co., Inc. D.B.A. The Genesis Group and Phil Burks make no representations or warranties with respect to the contents or fitness for a particular purpose beyond the cost of the software paid by the end-user.

The software contains valuable trade secrets and proprietary information. Unauthorized use of the manual or software can result in civil damages and criminal prosecution. As an end user, you agree to abide by and heed these statements.

License

Title to the media on which the program is recorded and to the documentation in support of the product is transferred to you, but title to the program, and all subsequent copies of the program, despite the form or media in or on license is not a sale of the original or any subsequent copy. You assume responsibility for the selection of the program to achieve your intended results, and for the installation, use, and results obtained from the program.

Support

Customer satisfaction is our number one priority at Genesis. We are here to provide you with the best software possible, and we want to know when you have any questions, concerns or problems with GADI so that we can make it a better product for everyone.

Refer to the Troubleshooting & Support section of this manual for complete support and contact information.

Document History

Revision	Date	Description
Revision 1	March 2010	Initial Release
Revision 2	November 2011	Version 2.0 Release
Revision 3	April 2012	Version 2.1 Release
Revision 4	August 2012	Version 2.2 Release
Revision 5	March 2013	Version 2.2.2 Release
Revision 6	June 2013	Version 2.2.3 Release
Revision 7	October 2013	Version 2.2.4 Release
Revision 8	March 2014	Version 2.4.1 Release
Revision 9	November 2014	Version 2.4.3 Release
Revision 10	November 2014	Version 2.4.4 Release
Revision 11	January 2015	Version 2.4.5 Release

Table of Contents

About this Manual

Goals

This manual describes the role and function of the GADI solution.

Who Should Read This Manual?

This manual is written for the intended audience of novice to mid-level Motorola trunked radio system users and novice to mid-level PC users.

How This Manual Is Organized

This manual is organized as follows:

- **Overview**: Provides a brief, high-level look at the GADI solution.
- **Installation**: Provides instructions for the installation of the GADI solution.
- **Configuration**: Provides a guide to configuring the GADI solution.
- **Operation**: Provides a guide to each of the functions of the GADI solution.
- **Troubleshooting**: Provides a list of solutions to common issues.
- **Appendix A**: SNMP MIB definition for the GADI solution.

This manual contains the following images, used to indicate that a segment of text requires special attention:

- ⓘ **Additional Information**: Additional information is used to indicate shortcuts or tips.
- ⊗ **Warning**: Warnings are used to indicate possible problem areas. Such as a risk of data loss, or incorrect/unexpected function.

Chapter 1 Overview

This chapter provides a high-level overview of the GADI solution, its function, and its components

This chapter contains the following sections:

- **Purpose**
- **Components**
- **Network Diagram**

Purpose

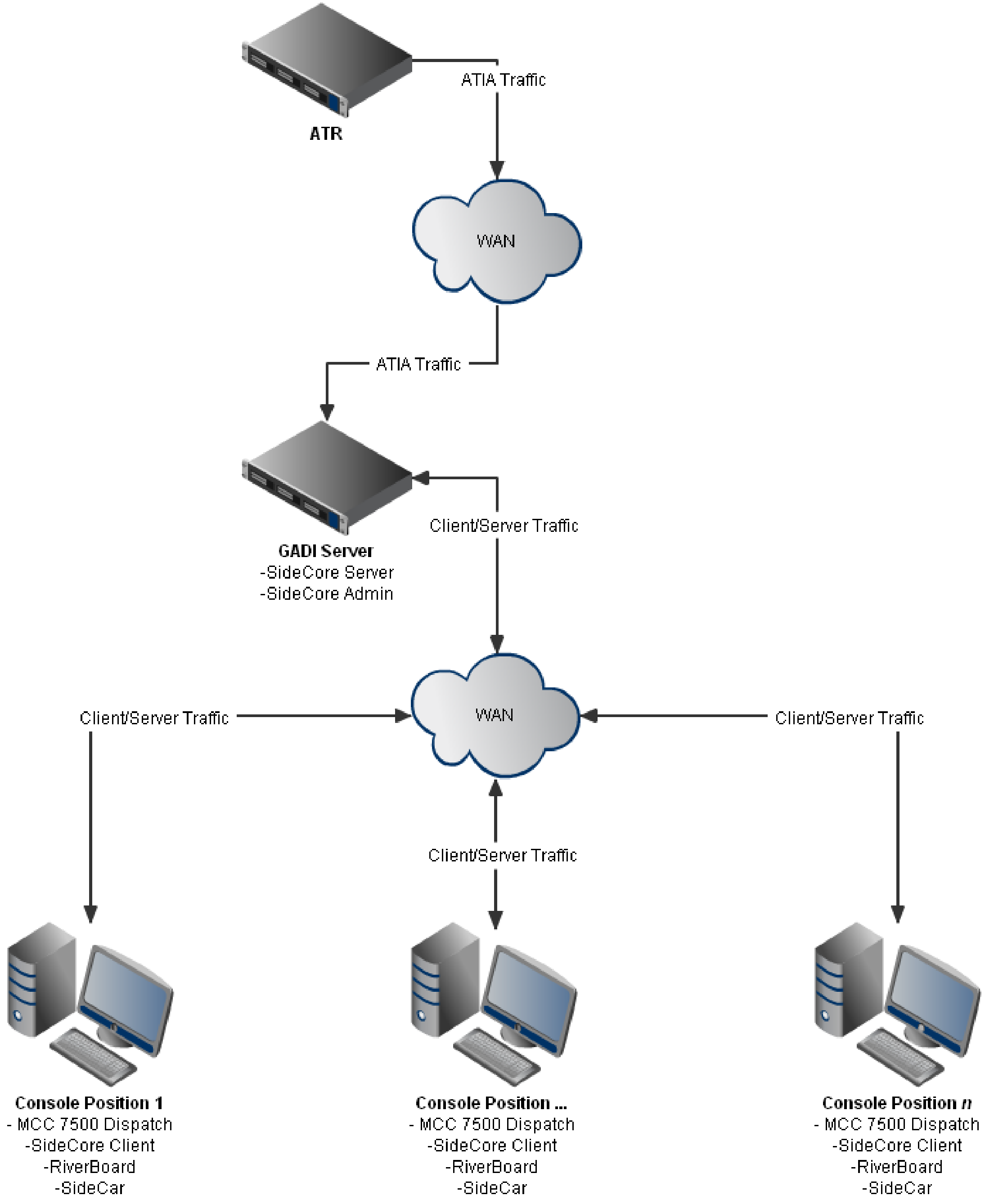
The purpose of the GADI solution is to provide enhanced functionality to a Motorola MCC 7500 dispatch console. The GADI solution runs in tandem with the MCC 7500 Elite software and provides additional features to the console operator.

Components

The GADI solution is made up of the following components:

- **SideCore Server:** This piece of software runs on a server machine to which all SideCore Clients have connectivity. The server software listens to the Motorola ATIA stream, manages settings for all client positions, and handles state management and routing of messages between clients.
- **SideCore Admin:** This piece of software is used to configure the SideCore Server and all SideCore Clients.
- **SideCore Client:** This piece of software runs on the MCC 7500 console machines and provides the graphical interface with which console operators interact on a daily basis.
- **Riverboard:** This is the main keyboard that provides console operators with the ability to perform GADI function via keystroke instead of using a mouse.
- **SideCar:** This is a supplemental keyboard that provides a numeric keypad and pointing device in the form of a trackball.
- **Footswitch:** This is a three-button footswitch allowing SideCore defined actions to be associated with each pedal.

Network Diagram



Chapter 2 Installation

The GADI solution is installed in three parts, in the following order:

1. SideCore Server(s)
2. SideCore Admin
3. Hardware (optional)
4. SideCore Clients

Installation is carried out by running the installation files for each piece of software, as provided by Genesis, and following the onscreen instructions. Additional notes for each piece are provided below:

Prerequisites

- All PCs with any piece of the GADI software installed on them must:
  - Be networked together
  - Be time synced via a central time source
  - Have .NET Framework 4.0 installed

SideCore Server

The SideCore Server software must be installed on a machine that has connectivity with all intended console positions. If a hot standby license is purchased, you may install the SideCore Server software on a second machine, providing redundancy to your clients.

ℹ The SideCore Server is installed as a Windows service, and once the installer completes, this service should start automatically. However, it is recommended that this be verified and that the service be started manually if it is not already running. This should be done prior to running the SideCore Admin software.

SideCore Admin

The SideCore Admin software can be installed on any machine that has connectivity with the primary server machine. For simplicity, it is recommended that the software be installed on the primary server machine itself. If your system was installed by Genesis, this is where it was installed.

The SideCore Admin software need simply be installed by running the provided installer and following the onscreen instructions, accepting the default values provided.

ℹ SideCore Admin must be run as an administrator.

Hardware

The optional GADI hardware components are USB devices that need simply be plugged into each MCC 7500 console position that the SideCore Client software will be installed on.

SideCore Client

The SideCore Client should be the last piece of the GADI solution to be installed. The installer should be run with default settings on all client machines.

ℹ The SideCore Client will not be able to connect to the SideCore Server until configuration is completed via SideCore Admin.

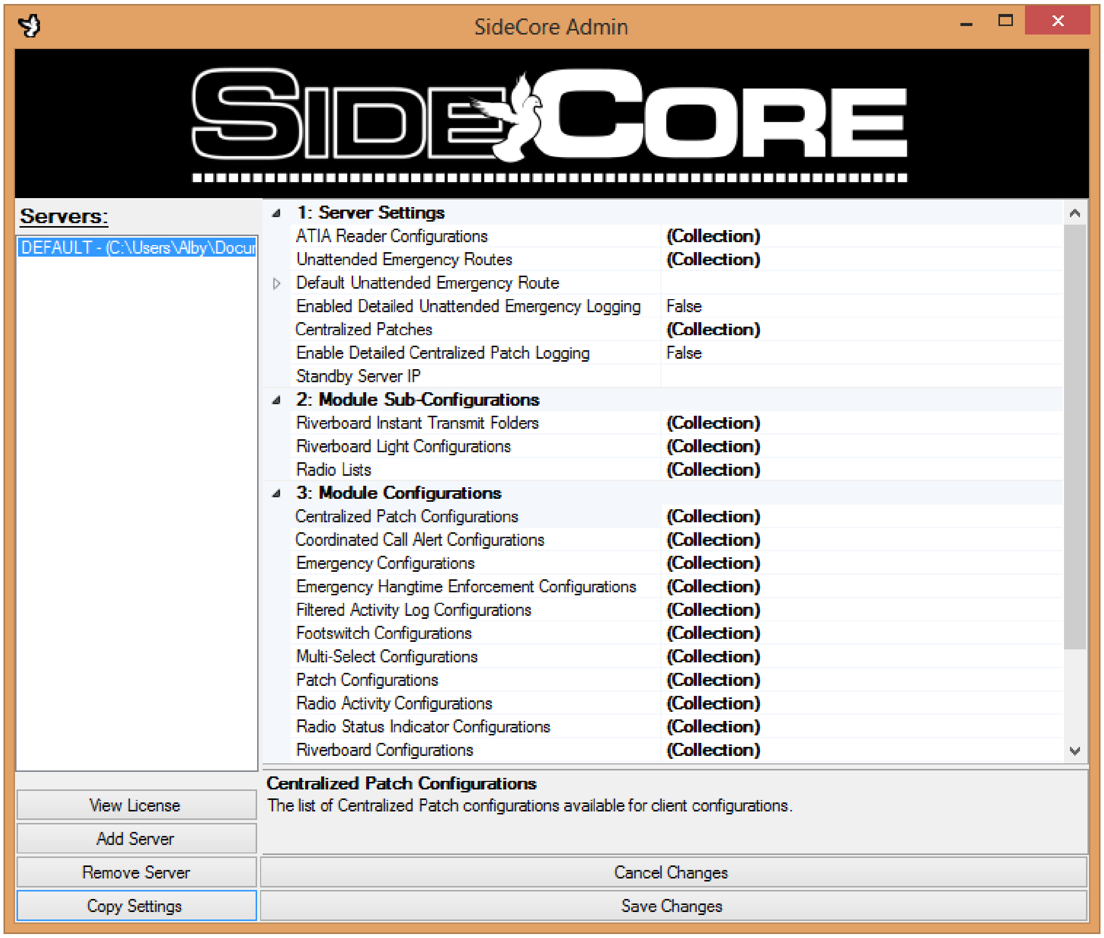
Chapter 3 Configuration

GADI configuration consists of the following three parts:

1. Server Settings
2. Centralized Client Configuration
3. Local Client Settings

SideCore Admin

The server settings are configured via SideCore Admin. Upon launching SideCore Admin, the following screen will be displayed:



1: Server Settings	
ATIA Reader Configurations	(Collection)
Unattended Emergency Routes	(Collection)
Default Unattended Emergency Route	
Enabled Detailed Unattended Emergency Logging	False
Centralized Patches	(Collection)
Enable Detailed Centralized Patch Logging	False
Standby Server IP	
2: Module Sub-Configurations	
Riverboard Instant Transmit Folders	(Collection)
Riverboard Light Configurations	(Collection)
Radio Lists	(Collection)
3: Module Configurations	
Centralized Patch Configurations	(Collection)
Coordinated Call Alert Configurations	(Collection)
Emergency Configurations	(Collection)
Emergency Hangtime Enforcement Configurations	(Collection)
Filtered Activity Log Configurations	(Collection)
Footswitch Configurations	(Collection)
Multi-Select Configurations	(Collection)
Patch Configurations	(Collection)
Radio Activity Configurations	(Collection)
Radio Status Indicator Configurations	(Collection)
Riverboard Configurations	(Collection)

Centralized Patch Configurations
The list of Centralized Patch configurations available for client configurations.

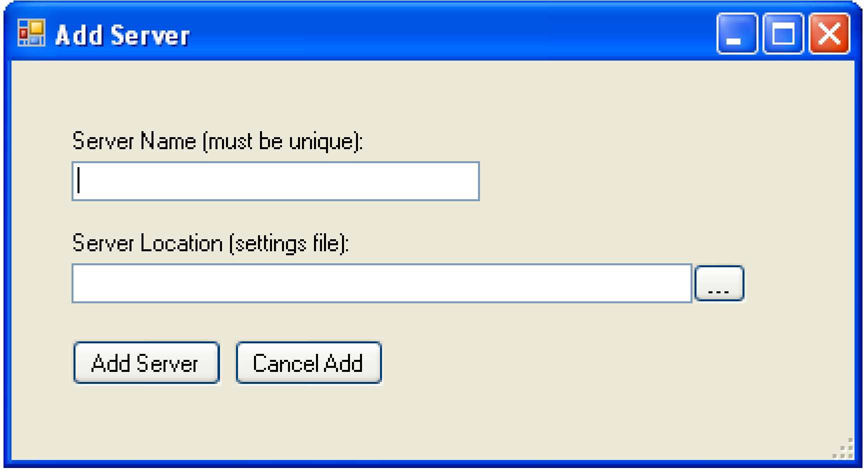
This is the main SideCore Admin screen and is where almost all GADI configurations occur. Each of the steps to configuring GADI is outlined below.

Adding and Removing Server Configurations

SideCore Admin allows you to manage multiple server configurations. Each configuration simply needs to be location in a different directory. This allows you to save different configurations for the system and quickly change between them as necessary. The configuration that is stored in the SideCore Server installation directory is the one that will be active in the system.

To Add a Server Configuration:

1. Click on the **Add Server** button at the bottom-left hand portion of the screen. This will open the Add Server screen:


Add Server
Server Name (must be unique):
Server Location (settings file):
...
Add Server
Cancel Add

2. Enter a unique name for the configuration.
3. Browse to the directory of the you are adding. If you are adding the active server configuration, it will usually be located under your "Program Files\Genesis\SideCore Server" directory.
4. Select the settings.dat file that is in the directory.
5. Click **Add Server**.

The new server configuration will now be available in the list of servers. Selecting it will allow you to edit its settings.

To Remove a Server

1. Select the server configuraiton in the list of servers on the left-hand side of the SideCore Admin screen.
2. Click the **Remove Server** button.
3. Confirm that you want to delete the server configuration when prompted.

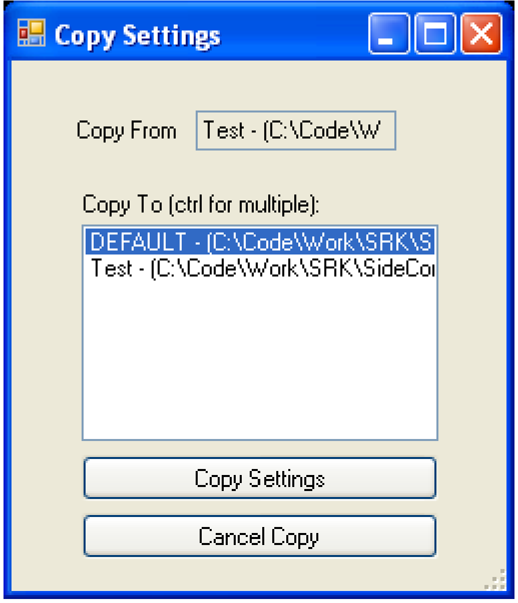
ℹ This will not delete the actual configuration file, it will simply remove it from the list of server configurations that are displayed in SideCore Admin.

ℹ It is only necessary to create a settings file for the Primary GADI server. The hot standby server will receive the configuration upon connecting to the primary. However, if the primary server is down for an extended period, you may add the hot standby server to the Admin in order to update its settings directly.

Copying Settings Between Configurations

The Admin server list will always contain a DEFAULT server configuration, which is not associated with the active server, but instead allows you to make a configuration without altering your current server configuration. To then copy the DEFAULT settings from one configuration to another, or vice versa, follow these steps:

1. Click on the server configuration you want to copy settings *from* in the list of servers.
2. Click the **Copy Settings** button, opening the copy settings screen:



Copy From: Test - (C:\Code\W

Copy To (ctrl for multiple):

DEFAULT - (C:\Code\Work\SRK\S
Test - (C:\Code\Work\SRK\SideCo

3. Select each of the server configurations you want to copy the settings to.
4. Click on the **Copy Settings** button.

The settings are copied to each of the server configurations you selected and automatically saved.

Editing Settings in SideCore Admin

Settings Sections

SideCore Admin displays all settings in the Settings Grid to the right of the server configuration list. The settings list is divided into sections, including:

1. Server Settings – Settings that pertain to the server.

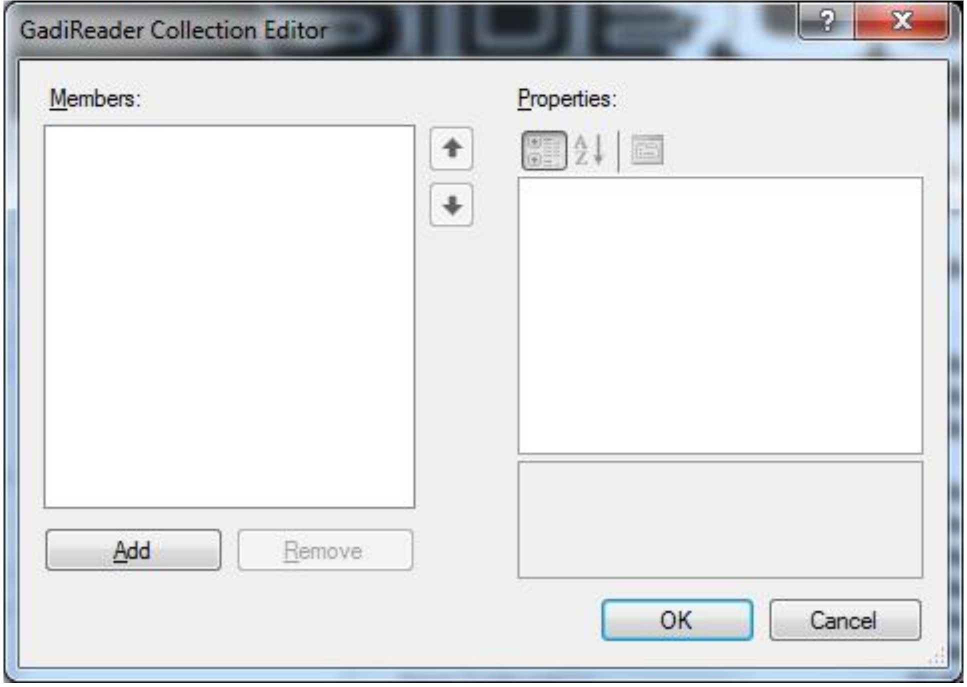
2. Module Sub-Configurations – Sets of configurations that are used as part of module configurations, such as a list of radio IDs.
3. Module Configurations – The settings for each SideCore Module.
4. Client Settings – The configuration of the clients that will be connecting to the server.

Using the Settings Grid

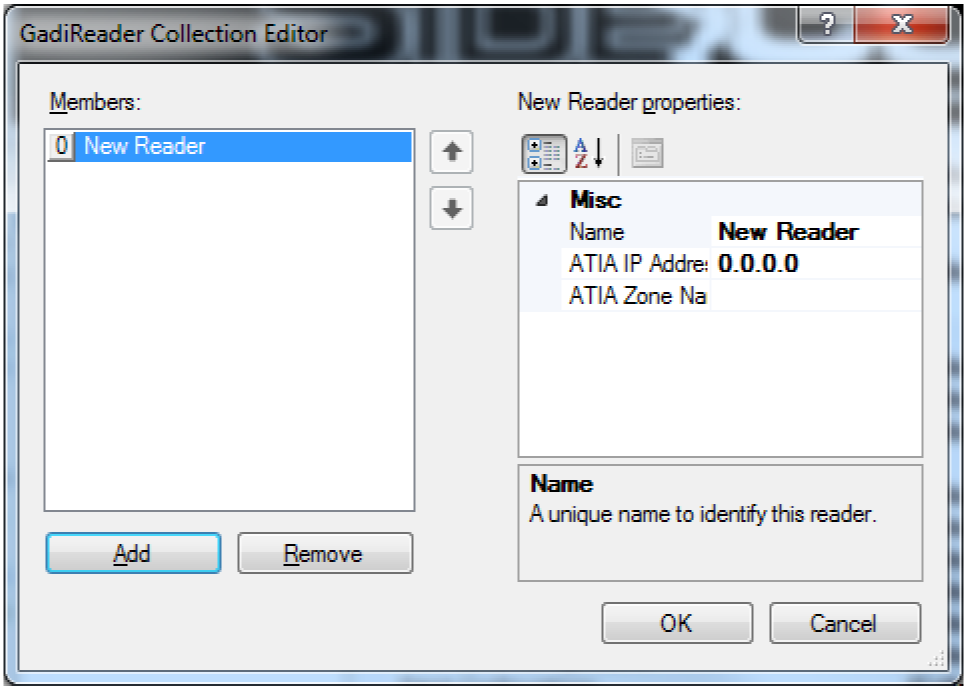
The settings grid consists of two columns: the left column which displays the name of the setting and the right column which displays the value.

Most settings in SideCore Admin are organized into collections of settings objects, such as a collection of ATIA Readers. Each reader would then contain specific properties that define how they should function.

You can identify a collection of settings by the **(Collection)** text that appears in the value column of the grid. To edit a collection of settings, simply click on the **(Collection)** box of the appropriate setting, and then click on the ellipsis button (…) that appears. This will open the Setting Collection Editor:



From here you can click **Add** to create a new entry, which will then select the new entry and display its properties:


GadiReader Collection Editor

Members:

0 New Reader

New Reader properties:

Misc

Name	New Reader
ATIA IP Addre:	0.0.0.0
ATIA Zone Na	

Name
A unique name to identify this reader.

From here you can now simply place your cursor within a property value box and change it as desired. You can click **Remove** to delete an entry, **Cancel** to close the box without keeping your changes, or **OK** to close and keep the changes.

NOTE: All settings objects that are part of a collection, such as a reader, will have a **Name** property that is used to reference the object. This is a user friendly name and does not affect the functionality of the system.

TIP: When editing settings for a collection containing many entries, it is possible to edit many at the same time by ctrl+click or shift+click or click and dragging to select multiple entries at a time.

Server Settings

ATIA Readers

ATIA Readers provide supplemental information to GADI via the ATIA stream from the Motorola ATR server.

Properties:

ATIA IP Address: the IP address of the GenSZAI Reader software. Refer to the GenSZAI Manual for directions on configuring a GenSZAI Reader.
ATIA Zone Name: the Zone Name of the ATIA data that you wish to listen to. This is configured via the GenSZAI Setup Utility. Refer to

the GenSZAI Manual for directions.

Unattended Emergency Routes

Unattended Emergency Routes define how the SideCore Server should route emergencies that are detected via the ATIA stream on resources that are not being monitored by any of its connected clients.

Properties:

Group: the Group Alias of the resource this route is for.
Clients: a list of the clients that should be notified if an unattended emergency occurs on this resource. After opening the Clients collection, hold ctrl to select multiple clients. Click anywhere outside of the list of clients to close the clients list.
Ignore: indicates whether emergencies that happen on this resource should be ignored (true) or forward to the selected clients (false).

Default Unattended Emergency Route

The default route is used when the SideCore Server goes through all of the routes, and does not find a connected client to forward the emergency to (assuming that the resource is not configured to be ignored). It has the same fields as a standard route, but only the Clients and Ignore settings are used.

ℹ If an unattended emergency still does not get forwarded to a client after the default route is applied, then it will be forwarded to all connected clients.

Enable Detailed Unattended Emergency Logging

Setting this option to true will cause the server to log all unattended emergency related events to the SideCore application log in the Windows Event Viewer.

Centralized Patches

If the Centralized Patch module is purchased, this list is used to define the patches that the server will manage:

Properties:

Permanent: indicates this patch should be automatically activated upon SideCore startup and re-established if ever de-activated.
Editable: indicates that the patch should be editable by clients.
Preferred Client: indicates the client that should be preferred to host the patch if more than one client is available to host it.

Configured Members: Clicking on this property will open a window where you enter one resource alias per line. This defines the initial members of this patch.

ℹ Configured members may not be removed from the patch by clients, even if the patch is set to editable.

Enable Detailed Centralized Patch Logging

Setting this option to true will cause the server to log all centralized patch related events to the SideCore application log in the Windows Event Viewer.

Standby Server IP

This is where the IP address of the hot standby server (if purchased) is specified.

ℹ Each server determines whether it is a primary or standby from the sidecore.ini file in its installation directory. If the file does not exist or does not contain the IP of a primary server, the server will run as a standby. If the file exists and contains a primary IP address (Primary=x.x.x.x), then it will run as a primary server.

Module Sub-Configurations

Riverboard Instant Transmit Folders

Instant Transmit Folders correspond to resource folders in the MCC 7500 Dispatch interface. They allow you to specify what resources correspond to each of the Instant TX buttons on the Riverboard. The Instant Transmit Folders option under Client Settings on the SideCore Admin screen allows you to define various different sets of resources, which can then be applied to Client Configurations.

Properties:

Instant x: indicates the resource that should be transmitted to when the console operator presses the corresponding Instant TX button on the Riverboard. Enter the *alias* of the resource desired.

Riverboard Light Configurations

Light Configurations allow you to define the behavior of the backlighting that is present on the Riverboard. Opening up the Light Configurations Collections brings up the following window:

Properties:

Rules: a collection of lighting rules which make up the lighting configuration. Consists of the following properties:

State: a state you want to the lights to change to. Options include:

- Off
- Solid Red
- Solid Green
- Solid Both
- Fast Flash Red
- Fast Flash Green
- Fast Flash Both
- Slow Flash Red
- Slow Flash Green
- Slow Flash Both
- Alternate

Keys: a list of keys whose backlighting should change to the selected state. Open the Keys collection and select the key(s) for which you want to change the backlighting (hold ctrl to select multiple), and then click anywhere outside of the list to close it.
TriggerEvent: an event that can occur in the system. When the selected event occurs on the system, the keys selected will have their backlighting changed to the state selected. Trigger events include:

- EmergencyAlarmReceived
- EmergencySilenced
- EmergencyAcknowledged
- EmergencyCleared
- Mute
- Unmute
- InstantTx1Start
- InstantTx1End
- InstantTx1Busy
- InstantTx1Pending
- InstantTx2Start
- InstantTx2End
- InstantTx2Busy
- InstantTx2Pending
- InstantTx3Start
- InstantTx3End
- InstantTx3Busy
- InstantTx3Pending
- InstantTx4Start

- InstantTx4End
- InstantTx4Busy
- InstantTx4Pending
- InstantTx5Start
- InstantTx5End
- InstantTx5Busy
- InstantTx5Pending
- InstantTx6Start
- InstantTx6End
- InstantTx6Busy
- InstantTx6Pending
- OutboundPrivateCallRinging
- InboundRadioPrivateCall
- InboundConsolePrivateCall
- PrivateCallStart
- PrivateCallEnd
- Patch1Active
- Patch1Inactive
- Patch1Empty
- Patch2Active
- Patch2Inactive
- Patch2Empty
- Patch3Active
- Patch3Inactive
- Patch3Empty
- Patch4Active
- Patch4Inactive
- Patch4Empty
- Patch5Active
- Patch5Inactive
- Patch5Empty
- TonePulseStart
- TonePulseEnd
- ToneWarbleStart
- ToneWarbleEnd
- ToneAlertStart
- ToneAlertEnd
- CallAlertInProgress
- CallAlertSuccessful
- CallAlertUnsuccessful
- GeneralTxStart
- GeneralTxEnd
- CallAlertAborted

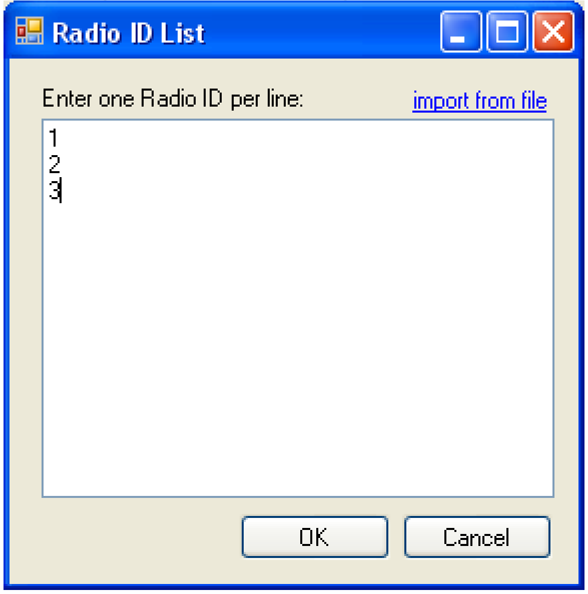
NOTE: Emergency events are NOT tied to specific emergencies. The events simply indicate that an emergency event has occurred in the system, not that all emergencies or any specific emergency is in a particular state. It is the responsibility of the customer to define light configurations that make sense. It is recommended that emergency key lighting be only used to indicate an emergency has come in, and then have it turned off on any other emergency events.

Duration: a number in milliseconds indicating how long the selected keys should remain in the selected state. When the duration ends, the key state will be set to Off. If you enter a 0 as the duration, the keys will remain in the selected state until it is changed by another rule.

Radio Lists

Radio Lists allow you define groups of radio IDs. These IDs can then be applied to Radio Filters for use with the GADI Filtered Activity Log.

Opening a Radio List brings up the following window:



Radio ID List

Enter one Radio ID per line:

1
2
3

Enter one Radio ID per line, as seen above. You may also click import from file to select a flat file of radio IDs to import. The file must contain one numeric ID per line.

Module Configurations

Module configurations are groups of settings that define how a specific module within SideCore should function. Multiple configurations can be created and later assigned to client configurations, allowing different clients to have modules that are configured differently.

Centralized Patch Configurations

The Centralized Patch module allows you to create patches that are managed by SideCore Server, enabling such features as: allowing patches to be edited from any console, creating permanent patches that automatically get activated and stay activated, and merging multiple patch groups together.

Properties:

Active Color: specifies the background color to be used for the patch buttons in the centralized patching window when a patch is in the active state.
Patch Member Indicator: specifies a font style to use for the name of the patch in the patch buttons in the centralized patching window when a patch has members (whether the patch is active or not).
Centralized Patch Host Count: indicates how many centralized patches that clients using this module configuration are allowed to host.
Centralized Patches: opens a drop-down list containing the Centralized Patches that were previously defined in the Server Settings. Hold ctrl and select the patches that clients using this module configuration should display on their screen.

Coordinated Call Alert Configurations

Coordinated Call Alert allows you to define lists of resources, typically fire stations, and their associated radio IDs. GADI can then coordinate alerting these stations and notifying all clients of their alert states.

Properties:

All Call Console: Indicates whether the client position should act as a dedicated All Call console. All Call consoles are used to play an audio file that alerts all stations through the paging encoder port of the console VPM. All Call consoles may not be used for any other functions.
UHF All Call Audio File Checksum: The MD5 hash of the UHF version of the All Call audio file. This is used to verify the integrity of the file before playing it on the system. This field may be left blank to skip verification.
VHF All Call Audio File Checksum: Same as above, except for the VHF audio file.
UHF Call Alert Resource: The ID or alias, preceded by '@', of the resource on which to send UHF call alerts.
UHF Call Alert Channel: The channel of the above resource to send UHF call alerts on (enter 0 if channel is not applicable to the resource).
VHF Call Alert Resource: Same as UHF.

VHF Call Alert Channel: Same as VHF.
Fire Stations: The list of stations to coordinate alerting for.
Fire Stations: a collection of fire stations, each consisting of the following properties:

> **Decimal UHF ID**: the decimal ID of the UHF radio for this station.
> **Decimal VHF ID**: the decimal ID of the VHF radio for this station.
> **Group Name**: indicates the name of the group of fire stations this station will belong to. The SideCore Client application will organize all stations based on this field. A button will be created for each unique group name among all stations, allowing the user to easily toggle the visibility of any stations that have the corresponding group name specified here.

Call Alert Duration: The length of time to consider a fire station engaged after a successful call alert (in seconds).
All Call Duration: The length of time to consider all stations engaged after a successful All Call transmission (in seconds).
All Call Audio Play Delay: How long the all call client should wait for the system to be ready before playing audio through the paging encoder port (in milliseconds).
All Call Audio Success Delay: How long the server should wait after a successful All Call audio transmission before notifying all clients of the all call engaged status (in milliseconds).
Call Alert Response Timeout: How much time the server should give each call alert to respond before assuming a failure (in seconds).
Status Message Time To Live: How long a status message should be considered valid. Messages received after this amount of time from their creation will be ignored.
Available Station Color: The background color to be used for stations in an available state.
Processing Color: The background color to be used for stations in a processing state.
Locally Engaged Color: The background color to be used for stations in an engaged state.
Station Busy Color: The background color to be used for stations in a busy state.
Station Selected Color: The background color to be used for stations in a selected state.
Remove From Queue On Failure: Indicates whether stations should be removed from the selected list upon a failed call alert.
Fire Station Width: Indicates how many pixels wide the graphical representation of each fire station should be.

Emergency Configurations

The Emergency module allows you to view all emergencies in one location within SideCore, and adds new features such as: unattended emergency handling, emergency volume management, and emergency tone management.

Properties:

Unattended Emergency Deassign Delay: a number in seconds indicating how long to wait before SideCore Client tries to deassign a resource that was assigned to a console as a result of an unattended emergency. If this number is set to 0, the resource will not be deassigned by SideCore Client.

Show Emergency Controls: indicates whether the emergency silence, acknowledge, and knockdown buttons should be display on the client.

Emergency Columns: defines the columns that appear in the emergencies list in the SideCore Client window. Column properties include:

Name: the column to be displayed.

Header Text: the text that should be shown in the header for this column.

Ordinal: indicates where the column should appear in the grid. Columns are ordered by ordinal from left to right, lowest ordinal number to highest ordinal number.

Width: how many pixels wide the column should be.

Selected Volume Management: indicates whether the SideCore Client should automatically lower the volume that is raised when emergencies happen on a selected resource.

Unselected Volume Management: indicates whether the SideCore Client should automatically lower the volume that is raised when emergencies happen on an unselected resource.

Unattended Volume Management: indicates whether the SideCore Client should automatically lower the volume that is raised when emergencies happen on a resource that is added due to an unattended emergency.

Default Volume Level: indicates the volume level that the SideCore Client should automatically lower the volume to if the volume is currently unknown.

Selected Tone Management: indicates whether the SideCore Client should stop the console generated emergency tone and play it separately when emergencies happen on a selected resource.

Unselected Tone Management: indicates whether the SideCore Client should stop the console generated emergency tone and play it separately when emergencies happen on an unselected resource.

Unattended Tone Management: indicates whether the SideCore Client should stop the console generated emergency tone and play it separately when emergencies happen on a resource that was added due to an unattended emergency.

Filtered Activity Log Configurations

Here you define the radio filters to be used by the Filtered Activity Log module.

Properties:

Show Instant TX: indicates whether an Instant TX button should be shown to the user. This button will allow the user to perform an Instant TX to the resource associated with the selected item in the activity log grid.

Show Emergencies: indicates whether emergency handling controls will be shown to the user. These controls will allow the user to handle any emergencies on the resource associated with the selected item in the activity log grid.

Radio Lists: opens a drop-down list where you can hold ctrl and click on the lists of radios to be included in this filter. The radio lists were defined previously under Module Sub-Configurations.

Include: indicates whether the filter should include the radios in the selected lists. Only activity for radios in the lists will be displayed. Otherwise, the radios in the selected lists will be filtered out and activity will be shown for all other radios.

Footswitch Configurations

Here you define the actions to be taken when each of the footswitch pedals is depressed.

Properties:

A Pedal Up and Pedal down property exists for each of the pedals: left, middle, and right. Each event, up and down, has the following two properties:

Action: the action to be taken by SideCore when the pedal event occurs. Currently available actions include (additional functions can be added on request):

- GeneralTXStart
- GeneralTXEnd
- MultiSelectAPBStart
- MultiSelectAPBEnd

Parameter: additional information needed to take the selected action. The parameter field is required for the following Actions:

MultiSelectAPBStart: The number of the MultiSelect group to start an APB on.

MultiSelectAPBEnd: The number of the MultiSelect group to end the APB on.

Multi-Select Configurations

Here you define the multi-select groups to be available in the Multi-Select module.

Properties:

Active Color: specifies the background color to be used for the multi-select buttons in the multi-select window when a multi-select group is in the active state.

Open Color: specifies the background color to be used for the multi-select buttons in the multi-select window when a multi-select is in the open state.

Multi-Select Member Indicator: specifies a font style to use for the name of the patch in the multi-select buttons in the multi-select window when a multi-select has members (whether the multi-select is active or not).

Multi-Select Groups: Clicking on this property will open up a collection editor where the specific multi-select groups are defined. Each group has the following properties:

> **Name**: the name to be displayed for this multi-select.
>
> **System Group Number:** (not-editable) the console defined number for this group (corresponds to the number displayed in Elite)
>
> **Enabled:** indicates whether this group will show up and be usable.
>
> **Locked:** indicates whether the dispatcher will be allowed to change the members of this group.
>
> **Group Members:** Clicking on this property will open a window where you enter one resource alias per line. This defines the list of initial members for this group.

Patch Configurations

Here you define the patch groups to be available in the Patch module.

Properties:

Active Color: specifies the background color to be used for the patch buttons in the centralized patching window when a patch is in the active state.

Open Color: specifies the background color to be used for the patch buttons in the centralized patching window when a patch is in the open state (only pertains to Multi-Select Linked Patches).

Patch Member Indicator: specifies a font style to use for the name of the patch in the patch buttons in the centralized patching window when a patch has members (whether the patch is active or not).

Show Patch Emergency Setup: specifies whether a patch emergency setup button will be visible to users. This button may be used to simultaneously setup a console-initiated emergency on all members of a patch group.

Patch Groups: Clicking on this property will open up a collection editor where the specific patch groups are defined. Each group has the following properties:

> **Name**: the name to be displayed for this patch.
>
> **System Group Number:** (not-editable) the console defined number for this group (corresponds to the number displayed in Elite)
>
> **Enabled:** indicates whether this group will show up and be usable.
>
> **Locked:** indicates whether the dispatcher will be allowed to change the members of this group.
>
> **Group Members:** Clicking on this property will open a window where you enter one resource alias per line. This defines the list of initial members for this group.
>
> **Multi-Select Linked:** Indicates whether the corresponding multi-select group (by system group number) should be automatically managed to match the state and members of this patch.
>
> **Show Multi-Select Open:** Indicates whether a button should be shown to allow the dispatcher to close the linked multi-select group when it is open.
>
> **Show Multi-Select APB:** Indicates whether a button should be shown to allow the dispatcher to transmit an APB over the patch linked multi-select group.

When a patch is Multi-Select Linked, the corresponding multi-select is automatically removed from the multi-select window and automatically managed behind the scenes to match the patch group. If all patches are configured to be multi-select linked, the multi-select window will not display any multi-selects. It is recommended that the multi-select window be removed in this configuration.

Radio Activity Configurations

Here you define the settings for Private/Radio Activity module.

Properties:

Private Alias: the alias of the private resource in the system. Private calls and Call Alerts will be sent via this resource.

Radio Activity Columns: defines the columns displayed in the Radio Activity window in the SideCore Client. Column properties include:

Name: the column to be displayed.
Header Text: the text that should be shown in the header for this column.
Ordinal: indicates where the column should appear in the grid. Columns are ordered by ordinal from left to right, lowest ordinal number to highest ordinal number.
Width: how many pixels wide the column should be.

The available columns to display include:

ID: The ID of the radio.
Alias: The alias of the radio.
Site: The site on which the activity occurred.
Zone: The zone on which the activity occurred.
LastConsoleStatus: An image indicating the last activity of the radio related to the console.
LastActivityTime: The time of the last action taken by the radio toward the console.
RadioStatusAlias: The alias of the radio status last received from the radio.
GroupID: The ID of the group the radio was last associated with.
GroupAlias: The Alias of the group the radio was last associated with.

Radio Status Indicator Configurations

Allow you to define relationships between radio statuses received from radios and colors. When a radio status is received for a radio from the system, the SideCore Client will display the associated color in the background of any radio id related grid cells.

Properties:

Radio Status Indicators: a collection of status-color relationships. Each member of the collection has the following properties:
Radio Status: the alias of the status to detect
Indicator Color: the color to set the id to display to when the status is received.

Riverboard Configurations

This is where the configurations related to the Riverboard hardware are selected.

Properties:

Auto Assign Instant TX Resources: when set to true, the SideCore Client application will automatically attempt to assign a resource to the

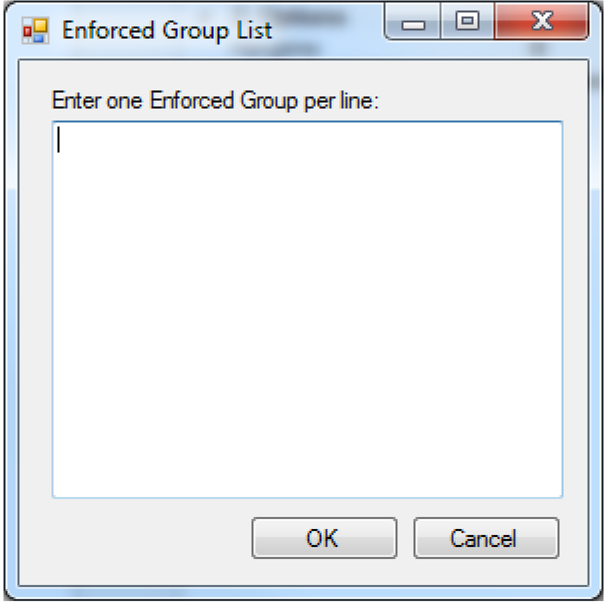
console if that resource is configured for an instant transmit button but is not found on the console position.
Use Elite Activity Log: when true, causes the Activity Log button to toggle the visibility of the Elite Acitivity Log window. When false, it will toggle the visibility of the SideCore Activity Log window.
Enable Key Event Logging: when true, all hardware key events are logged to the SideCore event log in the Windows Event Viewer.
Folder x: the Instant Transmit Folder configuration (previously created) that should be used by SideCore Client when the user is located on the associated MCC 7500 Dispatch resource folder. i.e. when the user is on MCC 7500 Dispatch resource folder 3, then SideCore Client will associate the resources configured under the Instant Transmit Folder configuration selected for Folder 3 here with the Instant TX buttons on the Riverboard.
Backlighting Configuration: the Riverboard Backlighting Configuration (previously created) that should be used for this module configuration.

Emergency Hangtime Enforcement Configurations

Allows you to configure groups to enforce emergency hangtime for. SideCore will automatically end any emergencies on the configured groups after a specified amount of time without any activity. If the system ends the emergency prior to the hangtime elapsing, SideCore will take no action.

Properties:

Hangtime: how long, in seconds, to wait after an emergency PTT before ending the emergency. Additional PTTs will reset the wait time.
Enforced Groups: opens the enforced group entry window:



One group alias is entered per line. Any entered groups will have their emergency hangtime enforced.

Client Settings

Client Configurations

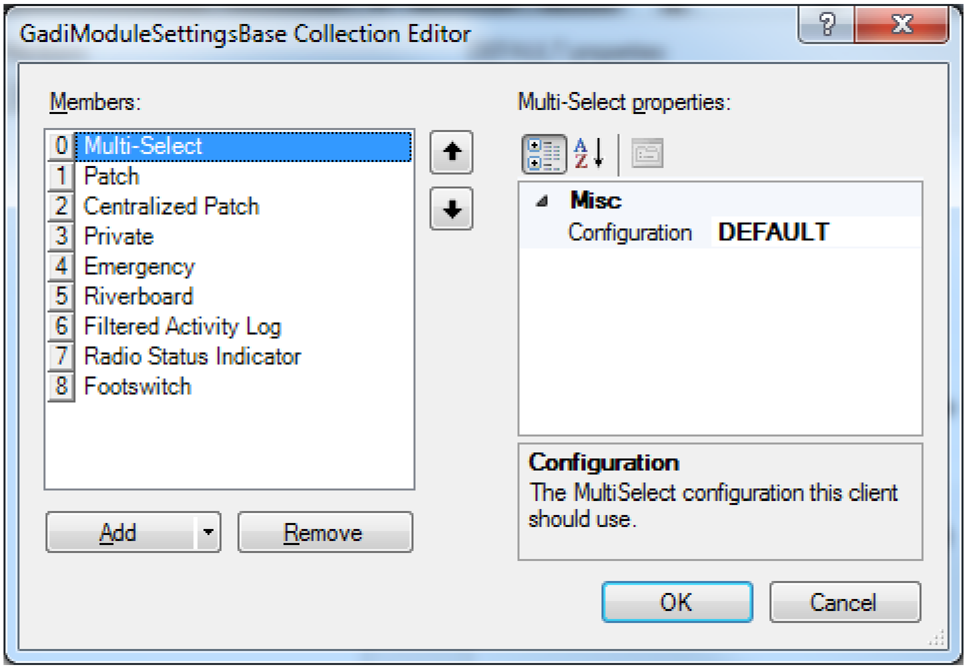
Client Configurations represent a complete set of Client settings. Multiple different configurations can be created and then applied to specific Client Connections.

Properties:

Font Size: indicates the size of the text for the SideCore Client window.
Show On Right Side: indicates whether the SideCore Client window should snap to the right-hand side of the user's screen. If set to false, it will snap to the left-hand side instead.
Form Width: indicates how many pixels wide the SideCore Client window should be.
Allow Client Customization: indicates whether end users should be allowed to customize their screen/module layout at runtime.
Notification Count: how many of the latest events should be displayed in the scrolling notification list at the bottom of the SideCore Client window.
Warning Color: the color that warning messages should be displayed in.
Error Color: the color that error messages should be displayed in.
Notice Color: the color that notice messages should be displayed in.
Log Critical Errors to Server: indicates whether the client should attempt to transmit any critical errors that are encountered to the server to be logged centrally. These errors are already logged locally; this option provides a central location for reviewing errors without having to go to each client position.

ℹ Even with this feature enabled, it is still possible that some errors may not be able to be transmitted to the server and would still need to be reviewed/retrieved directly from the client position.

Modules: the list of modules that should run on clients using this configuration. Opens the following screen:



Configuration DEFAULT

Configuration
The MultiSelect configuration this client should use.

Clicking the down arrow on the **Add** button enables you to select the module to add to this client. You then simply select the configuration to be used by the module on the right-hand side.

Clients

Clients represent specific MCC 7500 console positions that will be connecting to the server.

If the Centralized Patch module is being used, any client that will have their information changed must be offline when the changes are made. Failure to do so can lead to permanent patch instability.

Properties:

Name: the name of the machine that the SideCore Client software is installed on.

Clients identify themselves by providing their machine name to the server. If a machine name is changed, this field must be updated or the client will no longer be allowed to connect.

Radio ID: indicates the numeric ID of this console position.

Radio Alias: the alias of the client console in the radio system.
Configuration: the Client Configuration (previously created) that should be used at this console position. When a client connects from the IP address entered, it will be sent this configuration.
Enabled: indicates whether this console position is currently active in the system. If this is set to false, the server will not allow the client to connect.

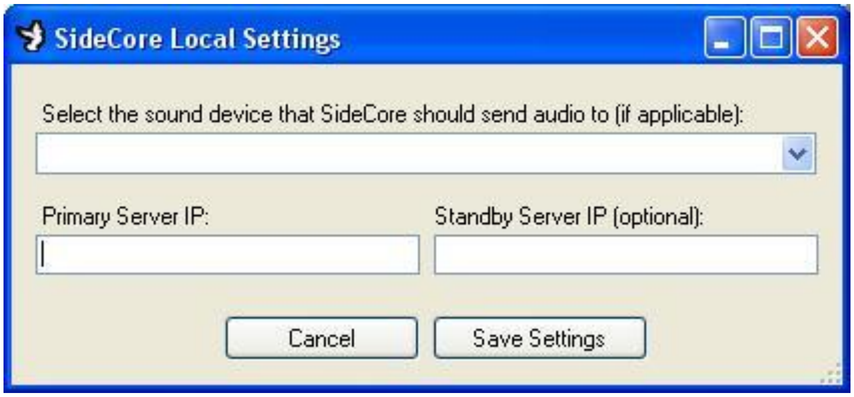
Finishing Up

When you are finished making changes for a server, click on the **Save Changes** button. The settings will be saved to disk, and the corresponding server will automatically load the settings and update any connected clients.

❌ If you make extensive changes to the Clients collection, specifically changing the Name field, it may be necessary to restart the SideCore Server. This is done by browsing the system services, and restarting the SideCore Server windows service.

Local Client Settings

When you start SideCore Client for the first time at a console position, you will be prompted with the following screen:

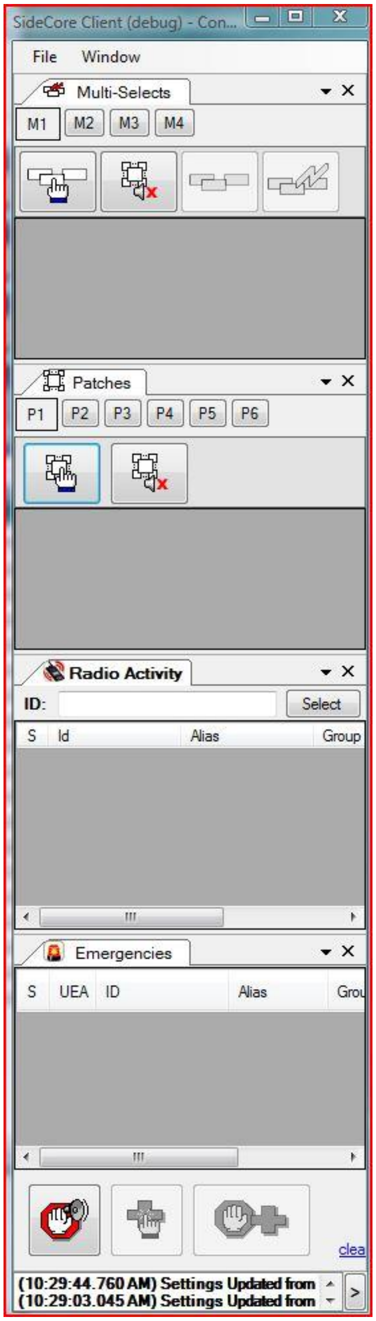

SideCore Local Settings
Select the sound device that SideCore should send audio to (if applicable):
Primary Server IP:
Standby Server IP (optional):
Cancel
Save Settings

The first entry specifies what sound device that SideCore Client should use to play any Emergency Tones (if configured to do so). Emergency tones will be heard via whatever speakers are connected to the sound device that is selected here.

The second entry specifies the IP address of the primary SideCore Server machine and the third specifies the IP address of the Standby SideCore Server machine (if any).

When there is a standby server configured for a SideCore Client, it will connect to both, but only listen to one at a time. If the primary connection fails, the standby will take over. It will then try to reconnect with the primary server every 30 seconds until it comes back online, at which point it will synchronize and return to listening to the primary.

This chapter will give an overview of the SideCore Client window, and will then describe each module of the GADI solution.



(10:29:44.760 AM) Settings Updated from
(10:29:03.045 AM) Settings Updated from

The SideCore Client Window

The SideCore Client window, seen left, is where the user-facing portions of the GADI modules reside.

Customizing the Layout

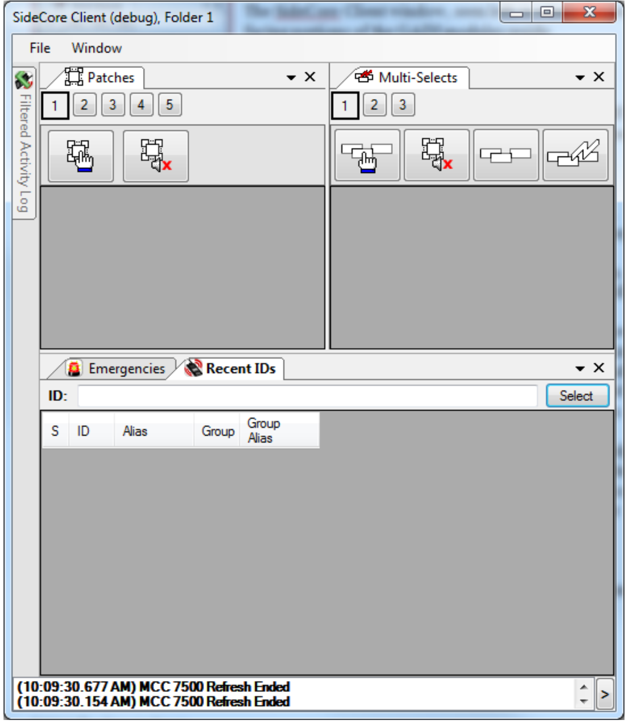
Each GADI module is contained within its own tab. If the client is configured to be allowed to customize its layout via SideCore Admin, the user will be allowed to customize the layout in several ways:

1. Floating: Double Clicking on the header of a tab, where the name of the module is displayed, will pop it out into its own window. Repeating this on the popped out tab will return it to its original location.

2. Tabbed: Clicking on a tab header and dragging will allow you to reposition the tab within the main SideCore Client window. Icons will appear to show you where you can then drop the module to reposition it. Dragging it on top of another module tab will allow you to make them share the same space in a tabbed fashion or side by side, depending on the icon you drop the module tab onto.

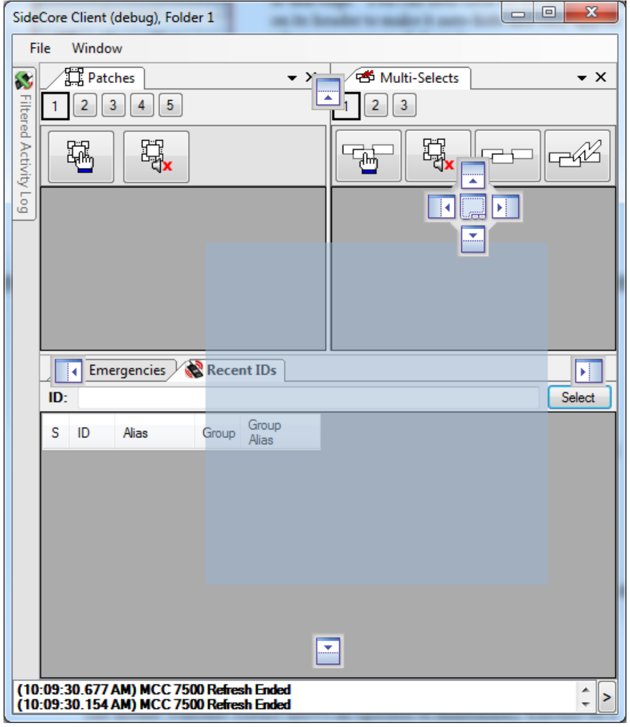
3. Docked: Clicking and dragging a module to the edge of the SideCore Client window will dock the module to that edge. You can then click on the push-pin icon on its header to make it auto-hide and only appear when you over or click on it.

An example of a customized window:



```
(10:09:30.677 AM) MCC 7500 Refresh Ended
(10:09:30.154 AM) MCC 7500 Refresh Ended
```

And customization guidance icons while dragging:



```
(10:09:30.677 AM) MCC 7500 Refresh Ended
(10:09:30.154 AM) MCC 7500 Refresh Ended
```

It is also possible to close windows by clicking on the X button of the tab header. Closed windows can later be opened by going to Window->Open in the File menu and selecting the desired module.

Startup Requirements

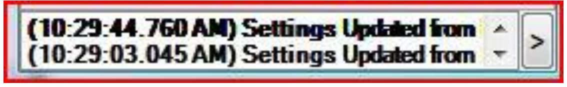
The SideCore Client Application requires Elite to be started in order to operate. If Elite is not started, a waiting for Elite message will be displayed until Elite is running, at which point the SideCore application will automatically finish starting.

If the SideCore Client application is an older version than the version of the SideCore Server, the user will be prompted with a message indicating that they need to upgrade. They will be given the option to continue running in offline mode. However, any server based features (unattended emergency, centralized patching, etc.) will not function until the client is upgraded to the server version.

ℹ If the Elite Activity Log window is open when the SideCore Client is started, the activity log will automatically be closed. The user may then simply re-open the log if it is needed.

Notifications

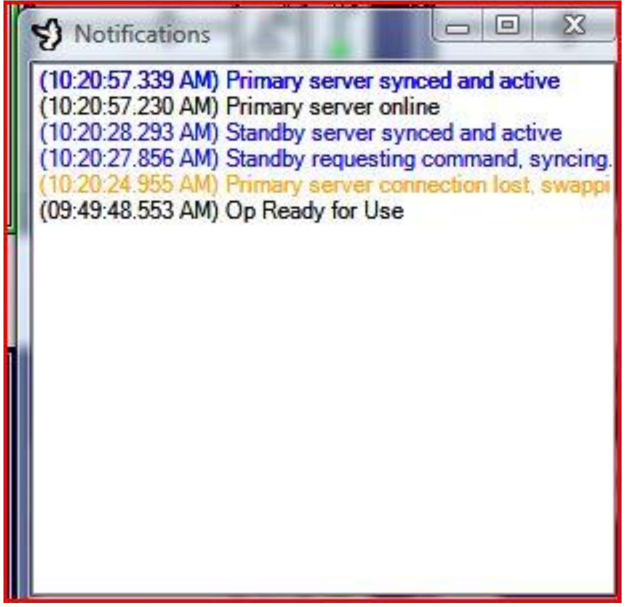
The bottom portion of the SideCore Client window displays a notification list.



(10:29:44.760 AM) Settings Updated from
(10:29:03.045 AM) Settings Updated from

This list displays the most recent events that have occurred in the system. The events are color coded based on their severity. The colors and number of events are both configurable via SideCore Admin.

Clicking on the > button to the right of the list will pop out a snapshot window that will allow you to easily scroll through the entire list of recent events:



(10:20:57.339 AM) Primary server synced and active
(10:20:57.230 AM) Primary server online
(10:20:28.293 AM) Standby server synced and active
(10:20:27.856 AM) Standby requesting command, syncing.
(10:20:24.955 AM) Primary server connection lost, swappi
(09:49:48.553 AM) Op Ready for Use

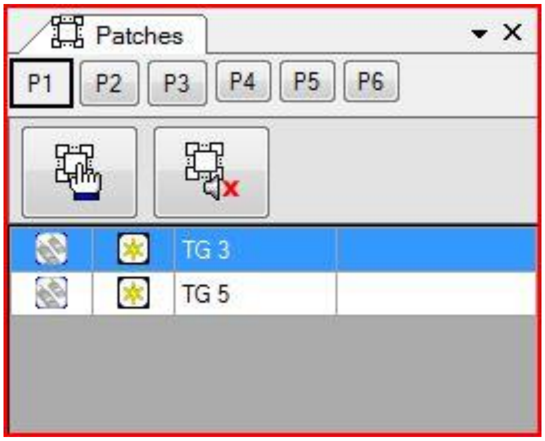
GADI Modules

Patch

The GADI Patch Module provides access to the patching functionality of the MCC7500 console. In addition to the functionality found in the Elite Patch window, GADI provides the following features:

- The ability to configure patch members without activating the patch.
- The ability to activate and populate pre-defined patches with one button click.
- The ability to define custom names for each patch.

Tab Overview



The top portion of the tab displays the **Patch List**. This list displays buttons with the names of each patch as configured via SideCore Admin.

Below the Patch List is the **Patch Action Pane**. This is where buttons that change the state of the patch are located.

The bottom area displays the **Patch Member** Grid. This grid displays the members of the patch that is currently selected and their current state. The columns of the grid, fr left to right, are:

1. Transmit status
2. Member status (added, pending, busy, removed, etc.)
3. Member alias
4. Owning console (if a member is pending due to use by another console position, the name of that console will appear here)

Viewing a Patch

Click on a patch button in the Patch List to bring the corresponding patch into view.

If there is an active patch when the SideCore Client is started that is not configured to be available to the user, SideCore will display the patch at the end of the patch list in brackets with a preceding X ([X16] would indicate patch 16 was active upon startup but not configured for the position). As soon as one of these patches is deactivated it will automatically be removed from the patch list.

Editing a Patch

Unlike Elite, the GADI Patch module allows you to edit patches without first activating them. This allows you to pre-define various patches and later activate them as desire with a single button click.

To add members to a patch:

1. Click on the first button in the Patch Action Pane. This is the Toggle Patch Edit button. The button will turn blue to indicate that the patch is in the editing state.
2. In Elite, find the resource you want to add to the patch and click on the Patch/Multi-Select toggle button that is configured on the resource.
   This button is added to the resource via Elite Admin and must be present in order to add members to patches via GADI. It will typically be configured with the help text of “Toggle Patch/Msel.”
3. The resource clicked on will be added to the current patch member grid. Click on the Toggle Patch Edit button again to end patch editing.

To remove members from a patch:

1. Click on the Toggle Patch Edit button to enter the editing state.
2. A red X will appear to the left of each patch member in the Patch Member Grid. Click on the X to remove the associated member.

   OR:

   In Elite, find the resource you want to remove from the patch and click on the Patch/Multi-Select toggle button that is configured on the resource.
3. Click the Toggle Patch Edit button again to end patch editing.

It is possible to edit a patch and a multi-select at the same time, by putting both the Patch and Multi-Select windows into edit state before adding/removing members. When editing both at the same time is not desired, the user must remember to take the Patch or Multi-Select window out of edit mode when they are done making changes.

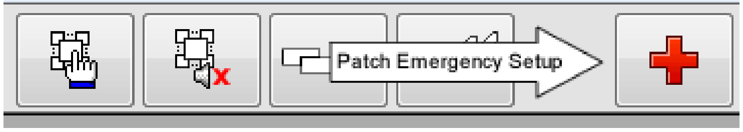
Activating/Deactivating a Patch

The second button in the Patch Action Pane is the Toggle Patch Active button. Click this button to activate a patch. The Toggle Patch Active button icon will change to indicate that the patch is active, and the background of the patch button in the Patch List will change color to indicate that it is active (color is configurable via SideCore Admin). Clicking the Toggle Patch Active button again will deactivate the patch and return the buttons to their original state.

If all members of the patch are currently in patches at other console positions, the patch will go into a pending state. While in the pending state, the Toggle Patch Active button will change to say 'cancel.' Clicking while in this state will cancel the activate operation and return the patch to inactive.

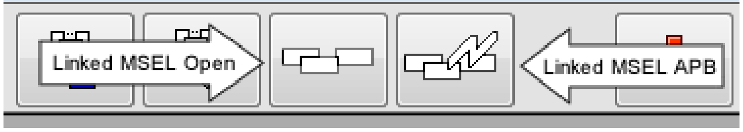
Patch Emergency Setup

The patch window can be configured by an administrator to display an emergency setup button. When clicked, this button will allow the user to simultaneously setup a console-initiated emergency on every member of an active patch.



Multi-Select Linked Patches

If a patch is configured to be multi-select linked (via SideCore Admin), the toggle multi-select open and multi-select APB transmit buttons may also be displayed.



Opening a Multi-Select Linked Patch

Opening a multi-select linked patch will actively select all of its members on the console. The multi-select will automatically be opened when the patch is activated or brought into view. To manually close or re-open the multi-select, click the third button in the Patch Action Pane (if configured to be shown). The button icon will change to indicate that the multi-select is open, and the button in the Multi-Select List will change color (color is configurable via SideCore Admin).

All Points Bulletin

An All Points Bulleting/APB can be sent to all members of an active or open multi-select linked patch by clicking and holding the last button in the Patch Action Pane (if configured to be shown). The transmission will last as long as the button is held down and the icon on the button will change to indicate the transmission is active. Releasing the button will end the APB.

Adding/Removing Members

Adding and removing members from a multi-select linked patch is handled in the same way as editing a patch. Multi-select members are automatically added/removed when the patch is edited.

ℹ If the dispatcher clicks on a resource in Elite while a multi-select linked patch is in the open state, GADI will automatically revert the action to prevent the patch from becoming out of sync with the linked multi-select. Members can only be added or removed by using the standard GADI patch editing procedure.

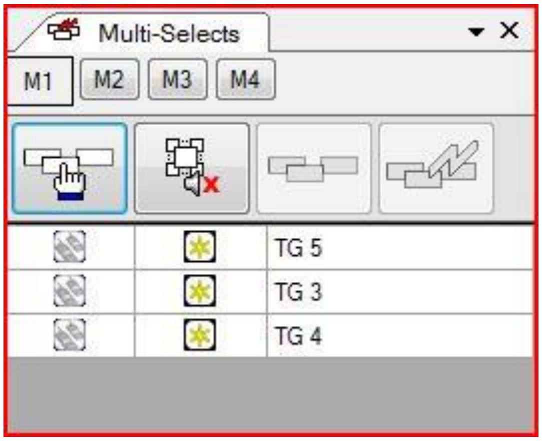
✖ The use of Motorola's Elite Dispatch software to patch or multi-select resources while the SideCore Client software is running with the patch and/or multi-select modules enabled is not supported. While possible, this usage can cause either application to show invalid patch/multi-select states and also cause the Motorola Elite Dispatch software to crash in certain scenarios. ***Elite should not be used for patching or multi-selecting while the SideCore Client is running with Patch Multi-Select linking enabled***.

Multi-Select

The GADI Multi-Select Module provides access to the multi-select functionality of the MCC7500 console. In addition to the functionality found in the Elite Multi-Select window, GADI provides the following features:

- The ability to edit multi-selects without activating (APB enabling) them.
- The ability to activate, populate, and open pre-defined multi-selects with a single button press.
- The ability to define custom names for each multi-select.

Tab Overview



The top portion of the tab displays the **Multi-Select List**. This list displays buttons with the names of each multi-select as configured via SideCore Admin.

Below the Patch List is the **Multi-Select Action Pane**. This is where buttons that change the state of the multi-select are located.

The bottom area displays the **Multi-Select Member** Grid. This grid displays the members of the multi-select that is currently selected and their current state. The columns of the grid, from left to right, are:

1. Transmit status
2. Member status (added, pending, busy, removed, etc.)
3. Member alias

Viewing a Multi-Select

Click on a multi-select button in the Multi-Select List to bring the corresponding multi-select into view.

Editing a Multi-Select

Unlike Elite, the GADI Multi-Select module allows you to edit multi-selects without first activating them. This allows you to pre-define various multi-selects and later activate them as needed with a single button click.

To add members to a multi-select:

1. Click on the first button in the Multi-Select Action Pane. This is the Toggle Multi-Select Edit button. The button will turn blue to indicate that the multi-select is in the editing state.
2. In Elite, find the resource you want to add to the multi-select and click on the Patch/Multi-Select toggle button that is configured on the resource.
   ℹ This button is added to the resource via Elite Admin and must be

present in order to add members to multi-selects via GADI. It will typically be configured with the help text of “Toggle Patch/Msel.”

3. The resource clicked on will be added to the multi-select member grid. Click on the Toggle Multi-Select Edit button again to end multi-select editing.

To remove members from a multi-select:

4. Click on the Toggle Multi-Select Edit button to enter the editing state.
5. A red X will appear to the left of each member in the Multi-Select Member Grid. Click on the X to remove the associated member.

   OR:

   In Elite, find the resource you want to remove and click on the Patch/Multi-Select toggle button that is configured on the resource.
6. Click the Toggle Multi-Select Edit button again to stop editing.

ℹ It is possible to edit a patch and a multi-select at the same time, by putting both the Patch and Multi-Select windows into edit state before adding/removing members. When editing both at the same time is not desired, the user must remember to take the Patch or Multi-Select window out of edit mode when they are done making changes.

Activating/Deactivating a Multi-Select

To enable APB transmissions to go out on all members of a Multi-Select, you must first activate the Multi-Select. To do this, click the second button in the Multi-Select Action Pane. The is the Toggle Multi-Select Active button. The icon on the button will change to indicate that the multi-select is active, and the background of the multi-select button in the Multi-Select List will change color to indicate it is active as well (color is configurable via SideCore Admin). Clicking the Toggle Multi-Select Active button again will deactivate the multi-select and return the buttons to their original state.

Opening a Multi-Select

Opening a multi-select will actively select all of its members on the console. To do this, click the third button in the Multi-Select Action Pane. The button icon will change to indicate that the multi-select is open, and the button in the Multi-Select List will change color (color is configurable via SideCore Admin). Clicking the button again will close the multi-select.

All Points Bulletin

An All Points Bulleting/APB can be sent to all members of an active or open multi-select by clicking and holding the last button in the Multi-Select Action

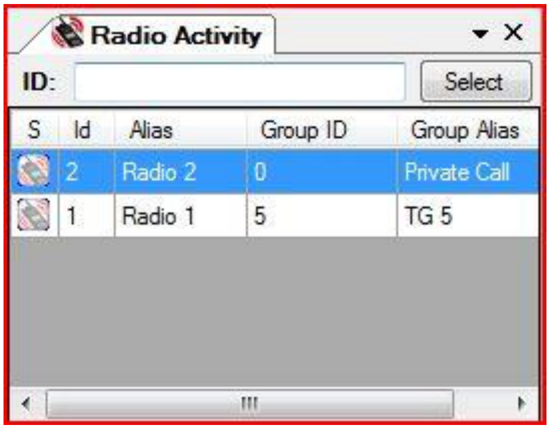
Pane. The transmission will last as long as the button is held down and the icon on the button will change to indicate the transmission is active. Releasing the button will end the APB.

❌ The use of Motorola’s Elite Dispatch software to patch or multi-select resources while the SideCore Client software is running with the patch and/or multi-select modules enabled is not supported. While possible, this usage can cause either application to show invalid patch/multi-select states and also cause the Motorola Elite Dispatch software to crash in certain scenarios. ***Elite should not be used for patching or multi-selecting while the SideCore Client is running***.

Private/Radio Activity

The GADI Private/Radio Activity module provides a way for the user to quickly find a list of the most recent radios that have interacted with the console directly, either outbound or inbound.

Tab Overview



S	Id	Alias	Group ID	Group Alias
	2	Radio 2	0	Private Call
	1	Radio 1	5	TG 5

The top part of the tab displays the **ID Entry Pane**, used to quickly find or add new IDs.

Below the ID Entry Pane is the **Recent Radio Grid**. This grid displays the last known status of each radio that has had interaction with the console. The columns displayed here are configurable via SideCore Admin. Right-clicking on a row in the grid will bring up all of the known details for the associated radio.

Adding/Selecting a Radio

The currently selected radio in the Recent Radio Grid will be the recipient of any direct radio related actions taken via GADI (Private Call, Call Alert). If a radio is not already present in the list or to quickly select a radio that is not currently visible, simply enter the ID of the radio into the textbox of the ID Entry Pane and click **Select** or press the Down Arrow. The radio will be added or selected and will be ready for interaction. You may also press Enter to add it to the list and immediately place a Private Call to the radio as well.

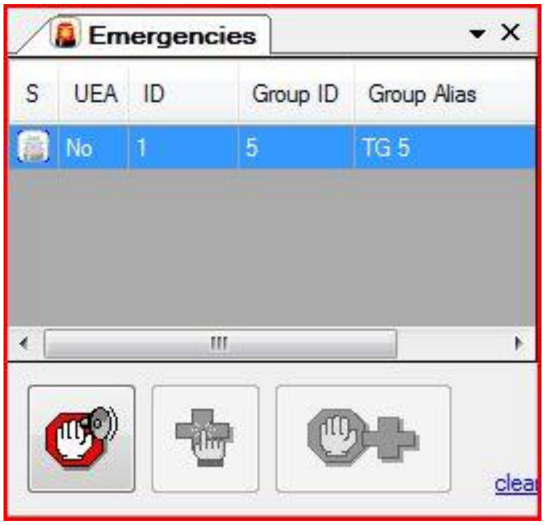
You may also simply click on a radio in the list to select it, or use the up, down, home, or end keys as well.

Emergency

The GADI Emergency module allows the user to view and interact with emergencies that occur on the system. The module provides the following features beyond that which is supplied via Elite:

- A single location for all emergency activity
- Prevent users from clearing an emergency they did not acknowledge
- Unattended emergency handling
- Custom emergency volume handling
- Custom emergency tone handling

Tab Overview



S	UEA	ID	Group ID	Group Alias
	No	1	5	TG 5

The top portion of the tab is the **Emergencies Grid**. This grid lists the emergencies on the system. The columns in the gird are configurable via SideCore Admin. Right-clicking on a row in the grid will bring up all known details about the emergency.

The bottom portion of the tab displays the **Emergency Action Pane**, which is used to interact with the selected emergency.

Silencing an Emergency

The first button in the Emergency Action Pane is the Silence Emergency button. Click this button to end all emergency tones at the local console.

Acknowledging an Emergency

The second button in the Emergency Action Pane is the Acknowledge Emergency button. Click this button to acknowledge the emergency that is currently selected in the Emergencies Grid.

Clearing an Emergency

The third button in the Emergency Action Pane is the Clear Emergency button. Once an emergency is acknowledged, click this button to clear the emergency.

Once an emergency is cleared, you can right-click on the emergency and select Remove Selected to remove it from the Emergencies Grid. You may also click the clear link to the right of the Clear Emergency Button to remove all emergencies that have been cleared from the Emergencies List.

ℹ GADI will not allow a console position to clear an emergency that it did not acknowledge.

Unattended Emergencies

If this option is purchased, GADI will monitor the ATIA data stream of the radio system to detect any emergency alarms that occur. When an emergency occurs, GADI will check to see if there are any consoles monitoring the associated resource. If there aren’t, it will then forward that emergency to any consoles that are configured in SideCore Admin to receive it.

When a console receives an unattended emergency, it is added to the emergency list and the background of the emergency window will flash red. The first console position to acknowledge the emergency will have the associated resource dynamically added to their console position. The resource will appear in the first available location within Elite. The emergency is then handled normally by the console operator.

❌ The dynamically assigned resource does NOT get populated to the Elite folder that is currently in view. It gets populated to the first available position in Elite. This may or may not be the current Elite folder. In this case, the dispatcher would need to change folder in order to see and interact with the resource.

The resource will then later be de-assigned automatically after the emergency is cleared. The time GADI waits before de-assigning the resource is configured in SideCore Admin.

The GADI unattended emergency feature makes a best effort to provide dispatchers with the ability to be notified of and handle emergencies that occur on non-monitored resources. The functionality is limited to detecting the emergency via the ATIA stream, alerting the dispatcher, and attempting to assign the resource to the console. Users of the unattended emergency functionality should be aware of the following considerations:

- We cannot guarantee the actual system status of the resource at the time of dynamic assignment. The resource may or may not still be in an emergency state from the radio system's perspective. As such, Elite may or may not display the resource in an emergency state.
- It is the responsibility of the system administrator to define unattended emergency routing that is valid and results in emergencies being routed to consoles that have the necessary security privileges to assign the resource the emergency occurred on. Failure to do this could interfere with proper unattended emergency functionality.
- Elite must be configured to allow dynamic assignments at the console. If this is not allowed, Elite will automatically de-assign any resources that GADI assigns to the console.

Emergency Volume Management

This optional feature allows an Administrator to instruct GADI to automatically revert the resource volume level of the emergency resource after Elite automatically raises it to the maximum when an emergency occurs. This can be set for Selected, Unselected, and Unattended emergencies independently.

Emergency Tone Management

This optional feature allows an Administrator to instruct GADI to automatically silence emergency tones that are played at the console when an emergency occurs and instead play them via a selected sound device on the local PC. This can be set for Selected, Unselected, and Unattended emergencies independently.

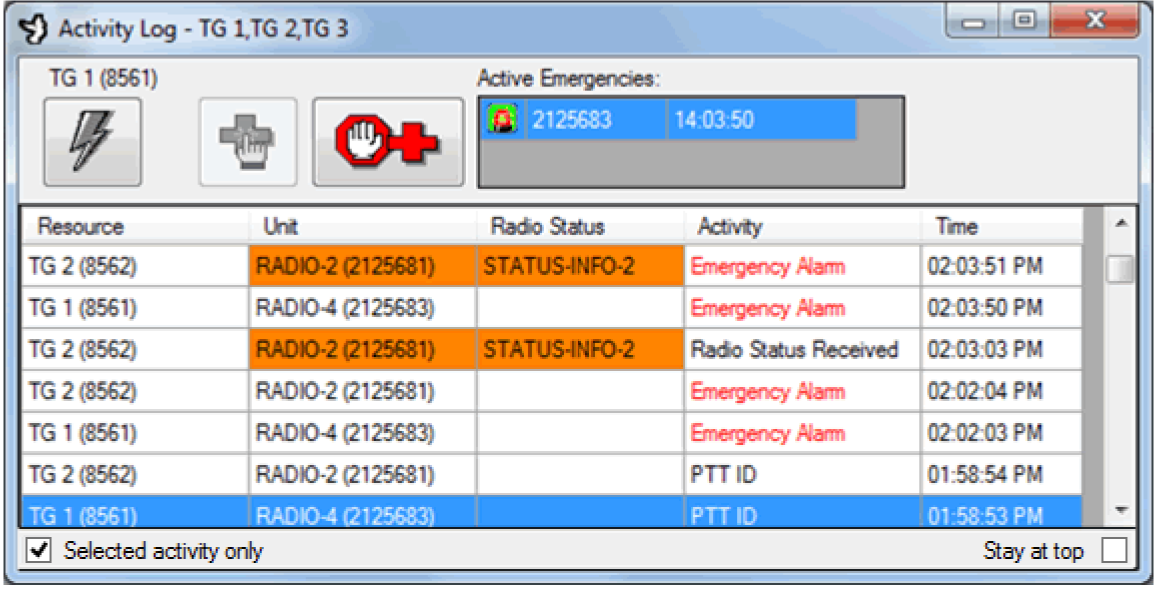
This feature was initially designed to allow dispatchers to have unselected emergency tones removed from their headsets.

Radio Status Indicator

The GADI Radio Status Indicator module provides a way for the console operator to quickly identify the last know status of a radio based on the last Radio Status message received from it.

There is no specific module tab for radio status indicators. Instead, the administrator sets up a relationship between statuses and colors via SideCore Admin, and then any grid cell within other module tabs that contains information related to a specific radio will display their background in the color corresponding to the last known Radio Status received by the radio.

An example status indicator from the Filtered Activity Log module:



2125683 14:03:50

Resource	Unit	Radio Status	Activity	Time
TG 2 (8562)	RADIO-2 (2125681)	STATUS-INFO-2	Emergency Alarm	02:03:51 PM
TG 1 (8561)	RADIO-4 (2125683)		Emergency Alarm	02:03:50 PM
TG 2 (8562)	RADIO-2 (2125681)	STATUS-INFO-2	Radio Status Received	02:03:03 PM
TG 2 (8562)	RADIO-2 (2125681)		Emergency Alarm	02:02:04 PM
TG 1 (8561)	RADIO-4 (2125683)		Emergency Alarm	02:02:03 PM
TG 2 (8562)	RADIO-2 (2125681)		PTT ID	01:58:54 PM
TG 1 (8561)	RADIO-4 (2125683)		PTT ID	01:58:53 PM

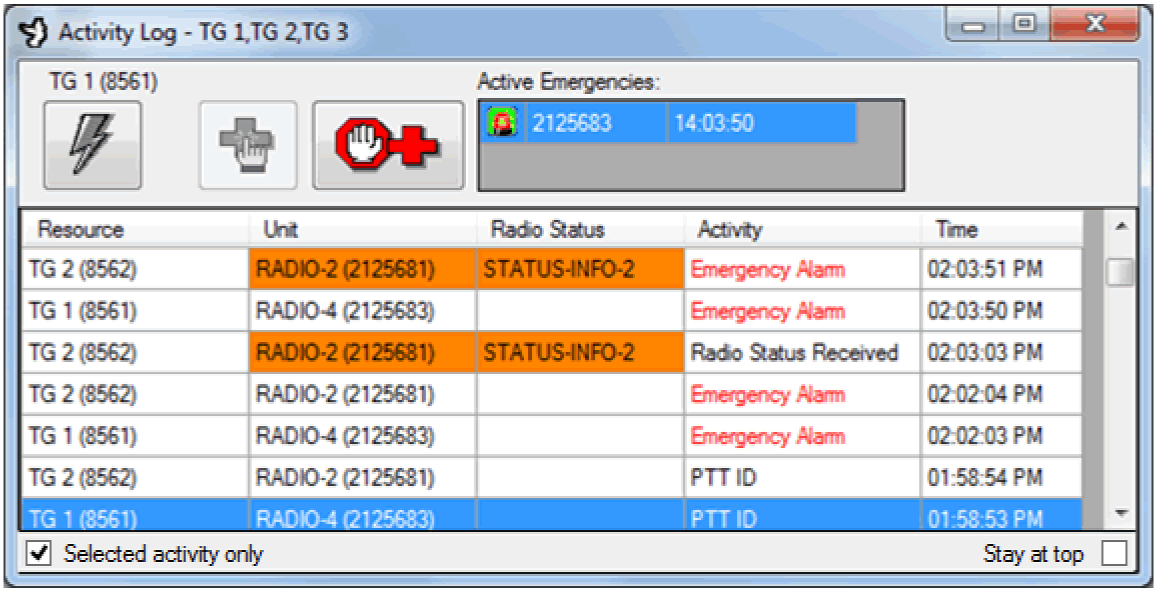
☑ Selected activity only

Stay at top □

Filtered Activity Log

This feature allows the user to display an activity log that is filtered based on the currently selected resource(s) and the radio ID filter configured for the position.

Window Overview



2125683	14:03:50

Resource	Unit	Radio Status	Activity	Time
TG 2 (8562)	RADIO-2 (2125681)	STATUS-INFO-2	Emergency Alarm	02:03:51 PM
TG 1 (8561)	RADIO-4 (2125683)		Emergency Alarm	02:03:50 PM
TG 2 (8562)	RADIO-2 (2125681)	STATUS-INFO-2	Radio Status Received	02:03:03 PM
TG 2 (8562)	RADIO-2 (2125681)		Emergency Alarm	02:02:04 PM
TG 1 (8561)	RADIO-4 (2125683)		Emergency Alarm	02:02:03 PM
TG 2 (8562)	RADIO-2 (2125681)		PTT ID	01:58:54 PM
TG 1 (8561)	RADIO-4 (2125683)		PTT ID	01:58:53 PM

☑ Selected activity only

Stay at top □

The top portion of the window is the **Selected Resource Action Pane**. This area provides buttons to instant transmit to and handle emergencies on the resource selected in the Recent Activity Grid. The emergency handling buttons interact with the currently selected emergency in the Active Emergencies list to the right. This pane can be configured to be shown or hidden and whether to display instant TX or emergency buttons or both.

If multiple resources are selected (via Multi-Select), the resource corresponding to whatever line is selected in the Recent Activity Grid will be the target of

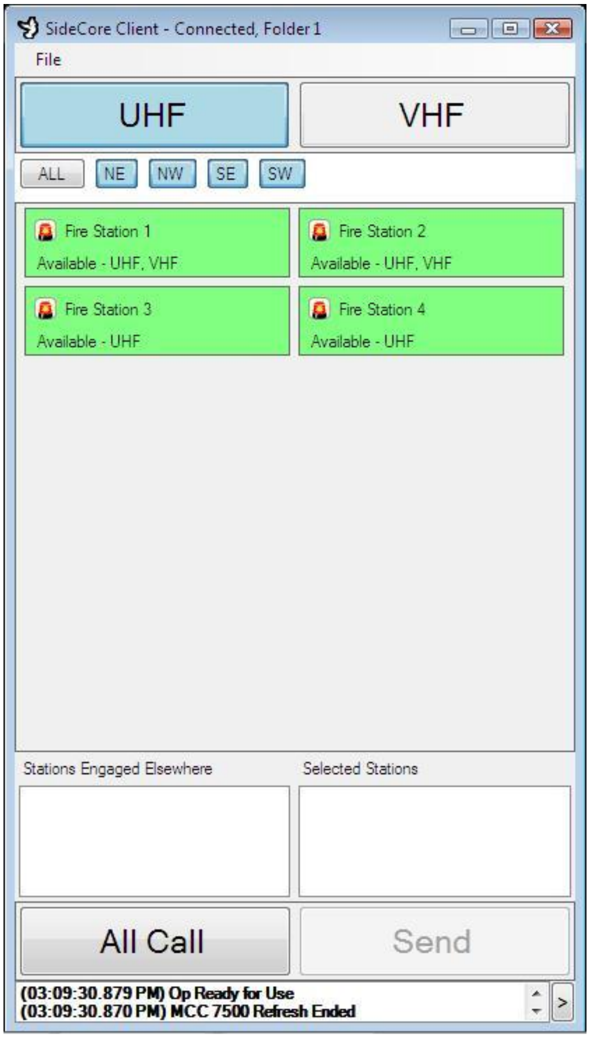
The middle portion is the **Recent Activity Grid**. This is where the most recent activity for the selected resources is displayed. As you change selected resources on the console, this list will change to reflect the activity for the currently selected resources.

At the very bottom of the window is an option to control how the Recent Activity Grid behaves. When the **Stay at** top option is selected, the grid will always return to the very top display new activity as it comes in. When unchecked, the grid will keep whatever line is selected visible as new activity comes in. When the **Selected activity only** option is selected, the grid will only display activity for the selected resources in Elite. When unselected, all activity will be displayed regardless of resource selection.

A radio ID filter can be defined within SideCore Admin and is used to exclude recent activity that occurs on certain radio IDs. To view the current radio

ID filter for a position, select **Show Radio Filter** from the main menu when the Filtered Activity Log window has focus (clicking anywhere in the window will bring it into focus).

Coordinated Call Alert



```
(03:09:30.879 PM) Op Ready for Use
(03:09:30.870 PM) MCC 7500 Refresh Ended
```

Mode Selection

Group Selection

Fire Station List

Engagements

Actions

Mode Selection

Description
This feature allows the user to select which resource/frequency to send station alerts on.

Action
Simply click on the button corresponding to the resource/frequency desired.

Result
When any call alert.or all call actions are performed, SideCore will direct them to the resource corresponding to the currently selected mode. If sending multiple call alerts at one time, all call alerts are sent via the mode selected at the time of the actual sending of the call alerts.

ℹ If a station does not have a radio configured for the selected mode, it will immediately fail upon send and be removed from the selected stations list.

Group Selection

Description
This feature allows the user to toggle the visibility of stations based on the groups they are configured in.

Action
Click on the button corresponding to the group of stations desired to be seen.

Result
All stations configured with the corresponding group name will become visible, and the group button background will turn blue. Clicking again will hide the stations and the button will return to gray. This allows the user to selectively view or hide groups of stations as the need arises.

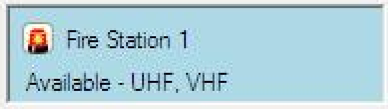
ℹ Clicking on the ALL button will toggle the visibility of all groups at once.

Fire Station List

Description
This list displays all fire stations that are in the currently selected fire station groups. This is where the user will select and view the current status of fire stations.

Action
Click on a fire station to be alerted.

Result
The station background color will change and the border will change to a depressed state to indicate it is selected.



Additionally, the station will be added to the Selected Stations list.

ℹ The selected background color is configurable via SideCore Admin.

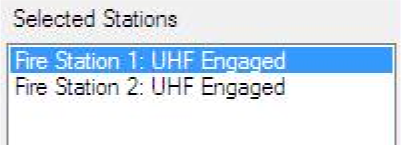
Engagements

Description

These lists provide a quick reference for what stations are current being interacted with, either at the local console, via the Selected Stations list, or via another console, via the Stations Engaged Elsewhere list.

Action
These lists simply display the currently active stations and there status in real time. The user does not directly interact with the lists.

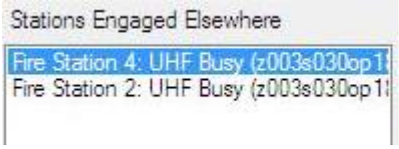
Result
The lists will update as station are interacted with locally:



Selected Stations

Fire Station 1: UHF Engaged
Fire Station 2: UHF Engaged

Or via other console positions:



Stations Engaged Elsewhere

Fire Station 4: UHF Busy (z003s030op1
Fire Station 2: UHF Busy (z003s030op1

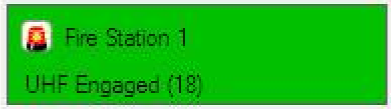
Actions – Call Alert

Description
The Call Alert feature allows the user to send individual call alert messages to one or more stations in a row.

Action
Select one or more stations in the Fire Stations List by clicking on them, then click Send.

Result
The Send button will become enabled after a station is selected and will display the number of selected stations. Upon clicking Send, SideCore will initiate sending call alerts to each selected station, sequentially. Each attempt will result in one of several states:

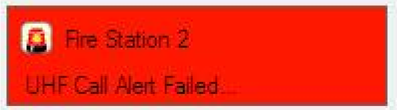
Successful



Fire Station 1
UHF Engaged (18)

Upon a successful alert, the countdown timer will begin for the station and it cannot be interacted with again until the timer expires.

Failed



Fire Station 2

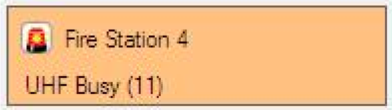
UHF Call Alert Failed...

Upon a failure, the station will flash red briefly and then be returned to the available state so that it can be interacted with again.

It is also possible to configure SideCore to keep failed stations in its queue so the user can easily attempt to resend alerts to them.

Busy

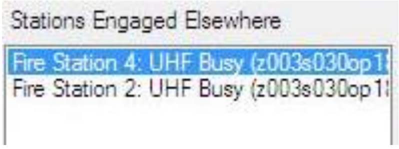
If the station happened to be reserved by another position prior to the local request, a busy status will be returned, and the countdown timer will be started.



Fire Station 4

UHF Busy (11)

The station will also be moved from the Selected Stations list to the Stations Engaged Elsewhere list.



Stations Engaged Elsewhere

Fire Station 4: UHF Busy (z003s030op1

Fire Station 2: UHF Busy (z003s030op1

All status background colors can be configured via SideCore Admin.

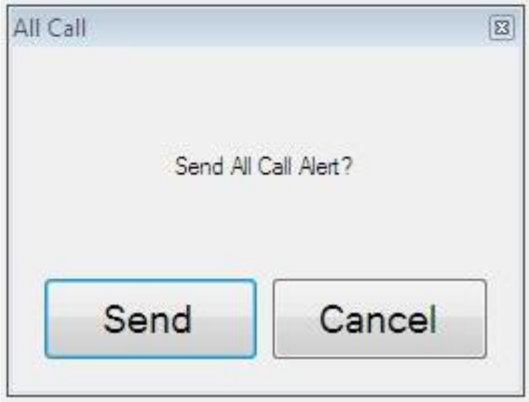
Actions – All Call

Description

The All Call feature allows the user to request that all stations be alerted at once.

Action

Click on the All Call button and confirm the All Call prompt:



All Call

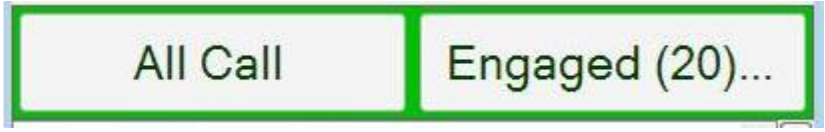
Send All Call Alert?

Result

SideCore will attempt to play the UHF or VHF (depending on the currently selected Mode) all call audio file via a dedicated All Call console.

The sound card of the all call console is wired to the paging encoder port of the console VPM, so that audio played via the all call console PC is then transmitted on the radio system.

All consoles then initiate an all call countdown timer and no alerting is possible until the timer expires.



ℹ Stations that did not initiate the All Call will show Busy instead of Engaged, though, as with the local position, they can still transmit to the stations via the MCC 7500 Elite application.

✖ If there are no All Call clients online, the All Call button will read “All call Offline” and it will be disabled.

Notifications

All Call

Description
Since All Call consoles are typically unmanned, all call notifications are provided to give adequate awareness when one of them goes offline.

Action
An All Call console goes offline.

Result
A popup window appears on all console positions indicating that the all call console has gone offline. If there is still an all call console online, the message will indicate that the all call service is still functional. If there are no more all call consoles online, the message will indicate that the service is offline completely, and the All Call button will become disabled.

Server Connectivity

Description
As the Coordinated Call Alert module depends upon server connectivity to provide coordination and alerting functionality, additional notifications are provided on server state changes.

Action
The current server goes offline.

Result
The entire coordinated call alert window will turn red and indicate that there is no server connectivity. If current server is the primary and a standby is active, the standby will momentarily take control and functionality will be restored. If the standby is active (i.e. primary is already offline), then the screen will remain deactivated until one of the servers comes back online.

All Call Window

The SideCore Client window on the All Call consoles consists entirely of a list of all call events that have occurred at the local position.

Centralized Patch

The GADI Centralized Patch module provides an alternative patching tab to the end user. This patching tab allows the user to access and interact with c*entralized patches* that are defined centrally and usable by any console.

Centralized patches are defined in SideCore Admin and can be configured to initially contain members or simply be empty and available to later add members to. Additionally each patch can be specified as:

- *Permanent*: the patch will be activated upon startup and monitored to ensure it is always activated. If the patch is closed for any reason, GADI will automatically re-establish it.
- *Editable:* indicates that end users can add or remove members from the patch. If this is set to false, users will only be allowed to activate and deactivate the patch.

  ℹ For editable patches, dispatch operators will only be able to remove members that were added by a dispatch operator; initially configured members are considered permanent.

Operation

From the dispatcher perspective, the Centralized Patch module looks and functions similarly to the Patch module with the following exceptions:

Patch Editing:
Unlike Patch and Multi-Select, when the user begins to edit either a local patch or a multi-select, Central Patch automatically stops editing. If it is desired to edit a multi-select or local patch at the same time as a Central Patch, the user must first put the local patch or multi-select window into edit mode and then put the central patch window into edit mode second.

Patch Activation:

The end user will still use the patch activate button to activate a patch, just as with the local patch module, but when done via central patching, the request will be sent to the patch server which will then look for a client that can host the patch. If no available clients are found, the user will get an error indicating this. If a host attempts to activate a patch and fails, the server will send it to the next available client. If a user requests to cancel the activation, the server will stop attempting clients after it receives the next client response. If the next client response is a successful activation, the user will need to deactivate the patch as the cancel action is no longer valid.

> ℹ Due to the nature of central patches, there may be increased delay between attempting to activate a patch and the patch becoming fully active. This delay will depend on how many clients are connected, what resources they have assigned, and how many clients have to be attempted before one succeeds.

Permanent Patches:
Permanent patches require no interaction from the end user. They are automatically set up by the server. The user will not be allowed to deactivate the patch and the server will attempt to re-establish the patch if the hosting client goes offline. Members may still be added to a permanent patch if the patch is defined as editable.

If any of the pre-configured members of a permanent patch are not active in the central patch, the patch button background color will show as orange to indicate to the user that it is not fully activated.

> ℹ If a SideCore Client that is hosting a Central Patch is taken offline, but the MCC 7500 Elite software remains operational at the position, the SideCore Server will attempt to move the Central Patch to a new position. However, since the underlying patch will still be active via the MCC 7500 Elite application, it will enter a pending state at the new client position. It will remain in this state until the patch is broken in Elite at the original console. While a central patch is in this state, any new members that are added to the patch will become active in the patch group, but will not be patched with the pending resources until the pending resources are removed from the patch at the original console and become fully active at the new position.
>
> The user will be notified of any hosted patches prior to closing the SideCore Client, however, it is recommended to always close Elite instead of SideCore when using the Centralized Patch feature.

Patch Members:

Members of an active patch will display the activating console and the hosting console instead of just the hosting console as in the local patch module.

> Since patches are not necessarily hosted on the console that activated them and are shared between consoles with different assigned resources, it is possible for there to be more members in a patch than are seen on a given console. *If a console does not have a resource assigned, it will not see that member as being part of the patch even if it is.*

Merging Patch Groups:
When it is possible to merge two patch groups (there are at least 2 patches that contain members and both are editable), a **Merge** button will appear on the Patch Action Pane. Clicking on this button will show a list of available patches to merge with. Simply click on a patch in the list to merge into it. The members of the current patch will be merged into the patch that was selected and the current patch will become disabled except for the merge button which will change to say **Unmerge**. Clicking Unmerge will undo the merge process. It is also possible to go to the patch that was merged into and unmerge from there as well. If multiple patches are merged into it, you will be prompted to select the patch you wish to unmerge.

Dropping Hosted Patches:
It is possible to drop all central patches that are being hosted at a given console position by going to Central Patches->Drop Hosted Patches from the file menu while your focus is in the central patch window. This function may be used to force patches to move from the console they are currently hosted on.

Footswitch

The GADI Footswitch module operates behind the scenes and has no end user display tab. When this module is purchased, a 3 pedal footswitch is provided at the console location. Each pedal of the footswitch is configured to perform a specific function via SideCore Admin. The dispatch operator then simply presses the appropriate pedal of the footswitch to perform the desired action on the system.

Actions currently possible include:

- General Transmit
- Multi-Select APB

Call Genesis to request additional functions.

Riverboard

The following is a description of all functions that can be performed via the GADI Riverboard. There is no specific module tab related to the Riverboard module.

ℹ The RiverBoard and SideCar should be plugged into the PC prior to starting the SideCore Client. If they are not, or if one is unplugged and plugged back in while SideCore is running, then the keyboards will not function. This can be resolved by going to *File->Re-Initialize Hardware*. Once this is done the keyboards will begin functioning.

General Transmit

Description
General transmit allows the operator to immediately transmit on any selected resources.

Action
Press and hold the **SELECT TX** button on the Riverboard.

Result
The console position will transmit on any selected resources for the duration that the **SELECT TX** button is held down.

Instant Transmit

Description
The Instant Transmit feature allows an operator to immediately transmit on a given resource, regardless of whether or not it is selected.

Action
Press and hold the **INST TX** button on the Riverboard that is configured for the resource that is desired to transmit to.

Result
The console position will transmit, on the resource that is configured via SideCore Admin for the current MCC 7500 resource folder and the pressed button, for the duration that the button is held down.

Activity Log

Description
This function allows the operator to open or close the MCC 7500 activity log window via a keystroke.

Action
Press the **ACTIVITY LOG** button on the Riverboard.

Result
If the MCC 7500 activity log window is open, it will close, otherwise it will open.

ℹ It is possible to configure this button to toggle the GADI Filtered Activity Log tab instead of the Elite Activity Log.

Private Call - Place

Description
This feature allows the operator to place a private call via the Riverboard/SideCar.

Action

1. Press the **PVT ID** button on the Riverboard to go to the ID entry box.
2. Enter a radio ID via the SideCar.
   OR
   Press the **DOWN** key on the SideCar to enter the radio IDs list and then use the **HOME**, **UP**, **DOWN**, or **END** keys to highlight a radio.
3. Press the **ENTER** key on the SideCar. If an ID is already selected in the radio IDs list, you can skip steps 1 and 2.

Result
A private call will be placed to the radio entered/selected. If a new radio was entered, it will be added to the radio IDs list and selected.

Private Call - Answer

Description
This feature allows the operator to answer any private calls that are placed to the console.

Action
Press the **INST TX** button on the Riverboard that is configured to correspond to the private resource via SideCore Admin.

Result
The call will be answered by the local console. This is analogous to picking up a telephone.

Private Call - Transmit

Description
This feature allows the operator to transmit to the private resource. This action is performed either after placing a call to a radio and having it respond or after pressing the **INST TX** button on the Riverboard that is configured to correspond to the private resource via SideCore Admin to answer an incoming call.

Action
Press and hold the **INST TX** button on the Riverboard that is configured to correspond to the private resource via SideCore Admin.

Result
The console will transmit on the private resource for the duration that the button is held down.

Private Call - End

Description
This feature allows the operator to end an active private call.

Action
Press the **PVT ID** button on the Riverboard.

Result
The currently active private call will end.

Call Alert

Description
This feature allows the operator to place a Call Alert to a given radio.

Action

1. Press the **PVT ID** button on the Riverboard to go to the ID entry box.
2. Enter a radio ID via the SideCar.
   OR
   Press the **DOWN** key on the SideCar to enter the radio IDs list and then use the **HOME**, **UP**, **DOWN**, or **END** keys to highlight a radio.
3. Press the **CALL ALERT** key on the Riverboard. If an ID is already selected in the radio IDs list, you can skip steps 1 and 2.

Result
A Call Alert is placed to the selected Radio.

Call Alert - Abort

Description
This feature allows the operator to abort a Call Alert that is in progress. A Call Alert may remain in progress while a radio is in private mode.

Action
Press the **CALL ALERT** button on the Riverboard.

Result

The currently active Call Alert call will be aborted, and the status icon in the radio list will be updated to reflect the current state.

Patch - View

Description
This feature allows to user to view a patch on the system, including any members and their statuses.

Action
Press the **PATCH X** key for the patch number to be viewed (if it is not already the currently viewed patch).

Result
The patch corresponding to the number of the key or button pressed will become visible, and the button on the SideCore Client window for that patch will become depressed (see button 1 above).

ℹ The 5 patch buttons on the Riverboard will correspond to the first 5 buttons in the patch window, in order. If you define more than 5, the additional patches will not be accessible via the Riverboard.

Patch - Activate/Deactivate

Description
This feature allows the operator to activate or deactivate a patch.

Action

1. Bring the desired patch into view.
2. Press the **PATCH X** key on the Riverboard to activate the patch.
3. Press the **PATCH X** key again to deactivate an active patch.

Result
The patch will become active (if it has members) after pressing the appropriate activate button. While the patch is active, the **Patch X Button** on the SideCore Client window's patch area will have a green background, and the **Activate/Deactivate Patch Button** will change to indicate that it is active.

When the appropriate buttons are pressed again, the patch will become deactivated.

Emergency - Silence

Description
This feature allows the operator to turn off any emergency alarm tones being played at the local console via a key on the Riverboard.

Action
Press the **EMERG SILENT** button on the Riverboard.

Result
The emergency tones sounding at the local console will end.

ℹ Pressing this button will also put the computer's focus to the emergencies list in the SideCore Client window. If at any point the operator needs to interact with an emergency and the focus is not in this window, the operator can simply press this button to be taken there.

Emergency - Acknowledge

Description
This feature allows the operator to acknowledge any outstanding emergency alarms via the Riverboard.

Action

1. Highlight the emergency to be acknowledged using the navigation keys or trackball on the SideCar.
2. Press the **EMERG ACK** button on the Riverboard.

Result
The highlighted emergency will be acknowledged by the local console, and the emergency status icon will turn green.

ℹ The operator on any console can then right-click the emergency to find out what console acknowledged it.

ℹ No other consoles will be allowed to clear this emergency

Emergency - Clear

Description
This feature allows the operator to clear an emergency that they previously acknowledged.

Action

1. Highlight the emergency to be cleared using the navigation keys or trackball on the SideCar.
2. Press the **EMERG CLR** button on the Riverboard.

Result

The highlighted emergency will be cleared by the local console, and the emergency status icon will gray out.

Alert Tones

Description
The alert tones feature allows the dispatch to instantly transmit a given alert tone on the selected resource via a key on the Riverboard.

Action
Press and hold one of the Riverboard alert tone keys: **ALERT, WARB,** or **PULSE**.

Result
The tone corresponding to the button pressed will be played for the duration that the button is held down.

Mute All

Description
This feature allows the operator to toggle the MCC 7500 All Mute option.

Action
Press the **MUTE ALL** button on the Riverboard.

Result
The state of the All Mute option on the console will be toggled. If it is was on it will be turned off, and if it was off it will be turned on.

Folder Select

Description
This feature allows the operator to change the current MCC 7500 resource folder that is being viewed.

Action

1. Ensure that the cursor is not currently in the ID Entry box in the SideCore Client window.
2. Press a **Number Key** (1-6) on the SideCar, corresponding to the number of the folder that is desired.

Result
The current MCC 7500 resource folder will be changed to the one corresponding to the number pressed.

ℹ In Windows 7 or later, SideCore must be started under a user account with equivalent privileges as is used to start Elite in order for folder select to function properly.

Backlighting

The RiverBoard includes a backlighting feature that can be used to show red or green lights beneath each of the keys. These lights are configured by an admin to be turned on or off based on system events. A user may enable or disable the backlighting feature by going to *File->Lights->Enable/Disable.*

Emergency Hangtime Enforcement

The Emergency Hangtime Enforcement module provides the ability to force the end of emergency calls on specified resources after a configurable hangtime period. This action is useful in scenarios where interference may be causing emergency calls to persist beyond the system configured hangtime, thereby tying up system resources unnecessarily (particularly on un-monitored talkgroups). In this scenario, SideCore would distinguish between interference and actual radio calls in order to end the emergency after the appropriate amount of hangtime.

If the system or a console operator ends the emergency prior to the hangtime elapsing, SideCore will take no action.

This chapter provides a guide to troubleshooting the GADI solution.

Logging

Each piece of the GADI solution logs all significant events and errors to the Windows Event Log. A custom log, named SideCore is created upon installation and configured to overwrite as needed, with a maximum log size of 10240KB and 10 days minimum retention.

In SideCore Admin there is a client setting to **Log Critical Errors to** Server. When this is enabled, in addition to logging locally, the client will attempt to communicate any critical errors to the SideCore server, where they are logged centrally in the SideCore event log on the server.

SNMP Messaging

The GADI solution provides SNMP traps to indicate when certain events or errors occur. These traps can be used to notify technicians when problems arise and to help diagnose problems in the system. All of these traps are originated from the SideCore Server software.

The following traps are provided:

- Reader Offline
- Reader Online
- No ATIA Data
- Reader Receiving ATIA Data
- Client Offline
- Client Online
- Settings Updated
- All Clients Online

The MIB definition for the GADI solution can be found in Appendix A.

Common Problems and Solutions

1. *Symptom*
   SideCore Client receives a connection denied message from the server

upon startup.

Cause
SideCore Client position is not configured properly in SideCore Admin.

Solution
Open SideCore Admin and ensure that, for the server in question, an entry exists in the Clients collection for the console in question. Ensure that the IP address is correct, the Radio ID is correct, and that the client is set to Enabled.

2. *Symptom*
   SideCore Client window does not appear after startup or does not connect to the server successfully, but no error message is received.

   Cause
   SideCore Client's local server setting is incorrect.

   Solution
   If SideCore Client window appears, go to File->Settings, and verify that IP address(es) entered is/are correct.

   If SideCore Client window fails to appear, browse to SideCore Client installation directory and manually open the settings.ini file with a text editor. Ensure that the first line contains the correct IP addresses.

3. *Symptom*
   Unattended Emergencies Offline notification is received at the console position.

   Cause
   The connected server is not receiving ATIA data from any of its connected ATIA readers.

   Solution
   Verify that ATIA packets are seen on the ATIA Reader machines using a utility such as Wireshark. Packets should be seen via UDP on port 8501. Contact a system administrator if they are not seen.

   If ATIA packets are being seen, open SideCore Admin and confirm that the ATIA readers for the server in question are properly configured. Ensure they have the correct IP addresses and that the Zone Name matches that which was defined in the GenSZAI6 Setup Utility.

   Consult GenSZAI documentation to troubleshoot ATIA readers if problems persist.

4. *Symptom*
   Emergency tones are not heard when they occur on a resource that is configured to be managed by GADI.

   Cause
   Sound device setting at local SideCore Client position is incorrect or speakers are not plugged into the selected device.

   Solution
   Go to File->Settings at the client position in question. Select the sound device that emergency tones should be sent to and click **Save**.

   Find the selected device on the PC and ensure that the speakers are plugged into the line out port.

5. *Symptom*
   After acknowledging an unattended emergency alarm, cannot find the resource associated with the emergency.

   Cause
   The resource was not added to the Elite folder that the user is currently viewing.

   Solution
   Elite will add the dynamically assigned resource to the first available position, which may or may not be in the folder the user is viewing.

   Switching to the first folder will typically show the resource.

6. *Symptom*
   Using buttons 1-6 on the SideCar does not change the folder in Elite.

   Cause
   The user's focus focus is in the Radio Activity Module's ID entry box.
   OR
   In Windows 7 or later, the user account used to start the SideCore Client is not the same as was used to start Elite.

   Solution
   If the user's focus is on the Radio Activity module's ID entry box, the 1-6 keys on the SideCar will enter numbers into this box instead of selecting a folder in Elite. Simply click outside of the entry box and the 1-6 will begin changing folders in Elite.

If the operating system is Windows 7 or later, the user account used to start SideCore must match the user account used to start Elite. Failure to do this will prevent the 1-6 keys from successfully changing folders in Elite. Starting both applications with the same user account will resolve the issue.

7. *Symptom*
   All buttons on RiverBoard or SideCar keyboards or footswitch do not work.

   Cause
   The USB hardware device was not registered with the PC USB driver when SideCore Client was started.

   Solution
   From the file menu, select Re-Initialize Hardware. If this fails, close SideCore Client and restart. If the device still doesn't function, unplug it from the PC USB port and then plug it back in and restart the SideCore Client again. If the device still doesn't function, reboot the machine. If this fails, attempt to use the device on another console position.

   If the device still fails to function after these steps, contact Genesis to send the device in for maintenance service.

8. *Symptom*
   Patch or Multi-Select group or member statuses do not match what is in the system.

   Cause
   This scenario can be caused by using Elite to perform patches and multi-selects while the SideCore Client is running.

   Solution
   While the SideCore Client makes a best effort to handle changes made to patches and multi-selects from other applications, this usage scenario is not fully supported and not recommended for regular usage.

   Restarting the SideCore Client will re-synchronize the statuses.

   If Elite was not used to perform any patch or multi-select actions, please contact Genesis to open a support ticket.

9. *Symptom*
   Elite crashes when using the SideCore Client patch->multi-select linking feature.

   Cause
   This scenario can be caused by using Elite to perform patches and multi-selects while the SideCore Client is running.

   Solution
   Elite may not be used to perform any patch or multi-select actions while the SideCore Client is running and has the patch->multi-select linking feature enabled.

   This usage will cause Elite to crash and require both application to be restarted.

Chapter 6 GADI SNMP MIB Definition

```
GADI-MIB DEFINITIONS ::= BEGIN

IMPORTS
        enterprises
                FROM RFC1155-SMI
        OBJECT-TYPE
                FROM RFC-1212
        DisplayString
                FROM RFC-1213;

genesis MODULE-IDENTITY
            LAST-UPDATED "200909010000Z"
            ORGANIZATION "Burks GenCore Co, Inc."
            CONTACT-INFO "The Genesis Group
                    601 Shelley Drive, Suite 202
                    Tyler, Texas 75701
                     www.GenesisWorld.Com
                     1 (877)548-0465 - US or Canada
                    +1 (903)561-6673 – International
                    +44 203 056-4382 - US offices from UK
                    +1 (903)561-6628 - Fax"
            DESCRIPTION
                        "Programs written by The Genesis Group for the Trunked Radio
industry."
            ::= { enterprises 32601 }

gadiMIB       OBJECT IDENTIFIER ::= {genesis 3}

gadiVersion OBJECT-TYPE
  SYNTAX  DisplayString
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
          "The version of GADI."
  ::= {gadiMIB 1}

-- Local Variables

gadiName OBJECT-TYPE
  SYNTAX  DisplayString
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
        "The variable tells the name associated with this GADI server."
  ::= {gadiMIB 2}

gadiIsPrimary OBJECT-TYPE
  SYNTAX  Integer32 { false (0), true (1)}
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
        "The variable tells whether this server is acting as the primary server."
  ::= {gadiMIB 3}

gadiConsoleClientsConnectedCount OBJECT-TYPE
  SYNTAX  Counter32
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
        "The variable tells the current number of connected console clients."
  ::= {gadiMIB 4}

-- The Readers

gadiReaderTable OBJECT-TYPE
            SYNTAX      SEQUENCE OF GadiReader
            MAX-ACCESS  not-accessible
            STATUS      current
            DESCRIPTION
                        "A table of readers configured for GADI."
    ::= { gadiMIB 5 }

gadiReader OBJECT-TYPE
            SYNTAX        GadiReader
            MAX-ACCESS          not-accessible
            STATUS        current
            DESCRIPTION
            "An entry used to select a GADI Reader."
            INDEX {gadiReaderIndex}
            ::= {gadiReaderTable 1}

GadiReader ::=
            SEQUENCE {
                        gadiReaderIndex         Integer32,
                                    gadiReaderName DisplayString,
            gadiReaderIpAddress  IpAddress,
            gadiReaderStatus        Integer32,
            gadiReaderPacketCount         Integer32
            }

            gadiReaderIndex OBJECT-TYPE
            SYNTAX      Integer32 (1..65535)
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                        "Index uniquely identifying a reader in this table."
    ::= { gadiReader 1 }

            gadiReaderName OBJECT-TYPE
            SYNTAX      DisplayString
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                        "The configured name for this reader."
    ::= { gadiReader 2 }

            gadiReaderIpAddress OBJECT-TYPE
            SYNTAX      IpAddress
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                        "The configured IP Address of this reader."
    ::= { gadiReader 3 }

            gadiReaderStatus OBJECT-TYPE
            SYNTAX      Integer32 {
                down (0),
                up (1),
                unknown (2)
               }
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                        "The current status of this reader."
    ::= { gadiReader 4 }

            gadiReaderPacketCount OBJECT-TYPE
            SYNTAX      Counter64
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
```

```
                        "The number of ATIA packets received since this reader came
online."
    ::= { gadiReader 5 }

-- The Clients

gadiClientTable OBJECT-TYPE
            SYNTAX      SEQUENCE OF GadiClient
            MAX-ACCESS  not-accessible
            STATUS      current
            DESCRIPTION
                        "A table of clients configured for GADI."
    ::= { gadiMIB 6 }

gadiClient OBJECT-TYPE
            SYNTAX        GadiClient
            MAX-ACCESS          not-accessible
            STATUS        current
            DESCRIPTION
            "An entry used to select a GADI Client."
            INDEX {gadiClientIndex}
            ::= {gadiClientTable 1}

GadiClient ::=
            SEQUENCE {
                        gadiClientIndex          Integer32,
                                    gadiClientName DisplayString,
                                    gadiClientRadioId       Integer32,
            gadiClientIpAddress   IpAddress,
            gadiClientStatus          Integer32
            }

            gadiClientIndex OBJECT-TYPE
            SYNTAX      Integer32 (1..65535)
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                        "Index uniquely identifying a client in this table."
    ::= { gadiClient 1 }

            gadiClientName OBJECT-TYPE
            SYNTAX      DisplayString
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
```

```
                "The configured name for this client."
   ::= { gadiClient 2 }

        gadiClientRadioId OBJECT-TYPE
        SYNTAX      Integer32
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
                "The radio id of the console associated with this client."
   ::= { gadiClient 3 }

        gadiClientIpAddress OBJECT-TYPE
        SYNTAX      IpAddress
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
                "The configured IP Address of this client."
   ::= { gadiClient 4 }

        gadiClientStatus OBJECT-TYPE
        SYNTAX      Integer32 {
             down (0),
             up (1),
             unknown (2)
            }
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
                "The current status of this client."
   ::= { gadiClient 5 }

-- Linked Server

gadiLinkedServer OBJECT-TYPE
   SYNTAX      GadiLinkedServer
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
        "The server this server is linked to."
   ::= { gadiMIB 7 }

GadiLinkedServer ::=
        SEQUENCE {
                gadiLinkedServerIpAddress   Integer32,
        gadiLinkedServerStatus        Integer32
```

```
            }

            gadiLinkedServerIpAddress OBJECT-TYPE
            SYNTAX      IpAddress
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                    "The configured IP Address of the server this server is linked to."
       ::= { gadiLinkedServer 1 }

            gadiLinkedServerStatus OBJECT-TYPE
            SYNTAX      Integer32 {
                 down (0),
                 secondary (1),
                 primary (2),
                                          unknown (3)
                }
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                    "The current status of this client."
       ::= { gadiLinkedServer 2 }

    -- Traps

    gadiReaderOffline TRAP-TYPE
            ENTERPRISE gadiMIB
            VARIABLES {
                    gadiReaderIpAddress,
                    gadiReaderName
                    }
            DESCRIPTION "GADI ATIA Reader Offline"
            ::= 0

    gadiReaderOnline TRAP-TYPE
            ENTERPRISE gadiMIB
            VARIABLES {
                    gadiReaderIpAddress,
                    gadiReaderName
                    }
            DESCRIPTION "GADI ATIA Reader Online"
            ::= 1

    gadiNoAtia TRAP-TYPE
            ENTERPRISE gadiMIB
```

```
        DESCRIPTION "No GADI Readers connected - no ATIA source."
        ::= 2

gadiReaderReceivingAtia TRAP-TYPE
        ENTERPRISE gadiMIB
        VARIABLES {
                gadiReaderIpAddress,
                gaidReaderName
                }
        DESCRIPTION "First ATIA packet received after reader connection."
        ::= 3

gadiClientOffline TRAP-TYPE
        ENTERPRISE gadiMIB
        VARIABLES {
                gadiClientRadioId,
                gadiClientName,
                gadiClientIpAddress
                }
        DESCRIPTION "GADI Client Offline"
        ::= 4

gadiClientOnline TRAP-TYPE
        ENTERPRISE gadiMIB
        VARIABLES {
                gadiClientRadioId,
                gadiClientName,
                gadiClientIpAddress
                }
        DESCRIPTION "GADI Client Online"
        ::= 5

gadiSettingsUpdated TRAP-TYPE
        ENTERPRISE gadiMIB
        DESCRIPTION "GADI Server Settings Updated"
        ::= 6

gadiClientsOnline TRAP-TYPE
        ENTERPRISE gadiMIB
        DESCRIPTION "All GADI Clients Online"
        ::= 5
END
```